Curriculum

Department of
Computer Science and Engineering

Undergraduates
Session: 2021-2022

Graduates
Session: 2021-2022

Shahjalal University of Science and Technology
Sylhet, Bangladesh

Published by:
Office of the Registrar
Shahjalal University of Science and Technology (SUST)
Sylhet 3114, Bangladesh

Syllabus Publication Committee:

Chairman:
Prof. Dr. Sabina Islam, Dept. of Statistics, SUST

Members:
Prof. Dr. Mushtaq Ahmed, Dept. of CEE, SUST
Prof. Dr. Sujoy Chakraborty, Dept. of Mathematics, SUST
Prof. Dr. Mohammad Muhshin Aziz Khan, Dept. of IPE, SUST
Prof. Dr. Mohammed Jahirul Islam, Dept. of CSE, SUST

Secretary:
Shaheena Sultana, Deputy Registrar, SUST

Cover Design:
Dept. of Architecture, SUST

Shahjalal University of Science and Technology
Sylhet-3114, Bangladesh
PABX: +88-0821-713491, 714479, 713850, 716123, 715393
Fax: +88-0821-715257, 725050
Website: www.sust.edu
E-mail: registrar@sust.edu

Contents

OVERVIEW OF THE UNIVERSITY AND DEPARTMENT

(At a glance)

Name of the University
Shahjalal University of Science and Technology, Sylhet

Establishment of the University
25 August 1986

Founder Vice Chancellor of the University
Professor Dr. Sadruddin Ahmed Chawdhury

Current Vice Chancellor of the University
Professor Farid Uddin Ahmed

First Academic Session of the University
1990-1991

Website of the University
www.sust.edu

E-mail of the University
registrar@sust.edu

Name of the Department
Computer Science & Engineering (CSE)

First Academic Session of the Department
1992-1993

Website of the Department
www.cse@sust.edu

E-mail of the Department
cse@sust.edu

PABX Extension of the Department
254

Founder Head of the Department
Professor Dr. M. A. Rakib

Current Head of the Department
Professor M. Abdulla-Al- Mumin

Programs Offering
BSc (Engg.), MSc, MPhil and PhD

FACULTY LIST (Current)

SL. No.	Full Name	Room No.	Cell Phone
	Professors		
1.	Dr. M. Shahidur Rahman	322	01914930807
2.	Dr. Md. Reza Selim	319	01972357830
3.	Dr. M. Jahirul Islam	321	01770348185
4.	Md. Abdullah Al Mumin, Ph.D.	312	01711445110
5.	Mr. Md. Masum	311	01919736248
6.	Dr. Farida Chowdhury	310	01743018917
7.	Dr. Md. Forhad Rabbi	323	01844175805
	Associate Professor		
8.	Mrs. Husne Ara Chowdhury	314	01712834619
9.	Mrs. Sadia Sultana	307	01911089612
	Assistant Professors		
10.	Mrs. Mahruba Sharmin Chowdhury	313	01917566699
11.	Ms. Ayesha Tasnim	308	01713328269
12.	Mr Md Eamin Rahman	212	01677014633
13.	Mr. Md. Saiful Islam*		
14.	Mrs. Marium-E- Jannat*		
15.	Mr Md Mahfuzur Rahaman*		
16.	Mr Md Mahadi Hasan Nahid*		
17.	Mr. Enamul Hassan	119	01914061632
18.	Mr. Moqsadur Rahman*		

19.	Mr. Summit Haque*		
20.	Arnab Sen Sharma*		
21.	Maruf Ahmed Mridul*		
	Lecturer		
22.	Mr. A. K. M. Fakhrul Hossain	211	01309354539

*On Study Leave.

Ordinance for Semester System for Bachelor's Degree

(This ordinance will replace other ordinances/resolutions etc. on the issues described here; however, it will not affect ordinances/resolutions on issues not mentioned here.)

1. Student Admission

1.1 Undergraduate Admission:
The admission committee of the university will conduct the admission process for Bachelor's degree as per the rules. The student will be admitted in the first semester of an academic year in the individual discipline of different schools. However the admission of foreign students will be subjected to the verification of academic records as per the university rule.

1.2 Student Status and Student Level:
Every student has to maintain his/her student status by getting admission paying necessary fees and register for required credits every semester. Unless a student graduate early by taking courses in advance, every student has to get admission in every semester successively. For book keeping purpose a student's level will be expressed by his/her year and semester. A student will be transferred to next level if he/she completes or appears in 80% of his designated courses at his/her present level. Once a student reaches 4th year 2nd (5th year 2nd for Architecture) semester he/she will be kept at this level until he/she graduates.

1.3 Re-Admission:
A student has to take re-admission if his/her student status is not maintained or one or more semesters were cancelled because of disciplinary action against him/her. In case of semester cancellation the student has to get re-admission in the same semester. The level (Year and Semester) of re-admission will be determined by his completed/appeared credits. A student will be eligible for re-admission in the first year first semester of the subsequent session if he/she was present in at least 25% of the classes of his/her major courses or appeared at the semester final examination and his/her admission/semester fees was clear in the past semester/session. Re-admitted students will always be assigned the original Registration Number.

1.4 Student's Advisor:
After admission every batch of student will be assigned to a student's Advisor from the teacher of his/her discipline to guide him/her through the semester system. Advisors will always be accessible to the students and will be ready to mentor them in their academic activities, career planning and if necessary, personal issues. There will be a prescribed guideline for the Advisors to follow.

2. Academic Calendar

2.1 Number of Semesters:
There will be two semesters in an academic year. The first semester will start on 1st January and end on 30th June, the Second semester will start on 1st July and end on 31st December. The routine of the final examination dates along with other academic deadlines will be announced in the academic calendar at the beginning of each semester.

2.2 Duration of Semesters:
The duration of each semester will be as follows:

Classes and Preparatory weeks	15 weeks
Final Examination	04 weeks
Total	19 weeks

These 19 weeks may not be contiguous to accommodate various holidays and the Recess before the final examination may coincide with holidays. The final grading will be completed before the beginning of the next semester.

3. Course Pattern

The entire Bachelor's degree program is covered through a set of theoretical, practical, project, viva and seminar courses. At the beginning of every academic session a short description of every available course will be published by the syllabus committee of each discipline.

3.1 Course Development:

3.1.1 Major and Non-Major Courses:
Syllabus committee of every discipline will develop all the courses that will be offered by that particular discipline and has to be approved by the respective school and the Academic Council. These include major courses for the respective discipline as well as non-major courses that will be offered to other disciplines. Non-major courses will be developed with close cooperation of the disciplines concerned keeping into consideration of the need of that discipline.

3.1.2 Syllabus:
(a) Major and Non-Major Courses: Syllabus committee will select and approve the courses from major courses of the discipline as well as non-major courses offered by other disciplines to complete the syllabus. The syllabus committee will also select a group of courses as core-courses and without these courses a student will not be allowed to graduate even if he completes the credit requirement. The committee may assign pre-requisite for any course if deemed necessary.
(b) Second Major Courses: The syllabus committee will select a set of courses of 28-36 credits from the major courses for a second major degree.

3.1.3 Course Instruction:
At the beginning of every semester the course instructor has to make a detailed plan of the course instruction in the prescribed form and supply it to the head of the discipline to make it available to the students. The course plan should have the information about the suggested text books, number of lectures per topic, number and type of assignments, number and approximate dates of mid-semester examinations

and mandatory office hours reserved for the students of the course offered. If not otherwise mentioned the medium of instruction is always English.

3.2 Course Identification System:
Each course is designated by a three-letter symbol for discipline abbreviation followed by a three-digit number to characterize the course. To avoid confusion new or modified courses should never be identified by reusing a discontinued course number

3.2.1 Discipline Identification:
The three-letter symbol will identify a discipline offering the course as follows. If same course is offered to more than one discipline, if necessary, an extra letter shown in the list may be used after the three digits to specify the department receiving the non-major course.

		School of Applied Sciences and Technology:	
1.	ARC	Architecture	A
2.	CEP	Chemical Engineering and Polymer Science	B
3.	CEE	Civil and Environmental Engineering	C
4.	CSE	Computer Science and Engineering	D
5.	EEE	Electrical and Electronic Engineering	E
6.	FET	Food Engineering and Tea Technology	F
7.	IPE	Industrial and Production Engineering	G
8.	MEE	Mechanical Engineering	Q
9.	PME	Petroleum and Mining Engineering	H
		School of Life Sciences:	
10.	BMB	Biochemistry and Molecular Biology	I
11.	GEB	Genetic Engineering and Biotechnology	J
		School of Physical Sciences:	
12.	CHE	Chemistry	K
13.	GEE	Geography and Environment	L
14.	MAT	Mathematics	M
15.	OCG	Oceanography	S
16.	PHY	Physics	N
17.	STA	Statistics	O
		School of Social Sciences:	
18.	ANP	Anthropology	a
19.	BNG	Bangla	b
20.	ECO	Economics	c
21.	ENG	English	d
22.	PSS	Political Studies	e
23.	PAD	Public Administration	f
24.	SCW	Social Work	g
25.	SOC	Sociology	h
		School of Agriculture and Mineral Sciences:	
26.	FES	Forestry and Environmental Science	P

		School of Management and Business Administration:	
27.	BUS	Business Administration	i
		Institute of Information and Coommunication Technology	
28.	SWE	Software Engineering	W

3.2.2 Course Number:
The three-digit number will be used as follows:
(a) First Digit: The first digit of the three digit number will correspond to the year intended for the course recipient.
(b) Second Digit: A discipline should use the number 0 and 1 for the second digit to identify non-major courses. The digits 2-9 are reserved for major courses to identify the different areas within a discipline.
(c) Third Digit: The third digit will be used to identify a course within a particular discipline. This digit can be used sequentially to indicate follow up courses. If possible even numbers will be used to identify laboratory courses.

3.2.3 Course Title and Credit:
Every course will have a short representative course title, declaration if it is core course, a number indicating the total credits as well as reference to prerequisite courses if any.

3.2.4 Theory and Lab Course:
If a single course has both Theory and Laboratory/Sessional part, then the course must be split into separate Theory and Lab courses and both should have separate course number. A student may not register for a lab course without registering or completing the corresponding theory course.

3.3 Assignment of Credits:

3.3.1 Theoretical:
One lecture per week (or 13 lectures in total) of 1 hour duration per semester will be considered as one credit. (There will be 10 minutes recess between theory classes). A theory course will have only integer number of credits.

3.3.2 Laboratory Classes:
Minimum two contact hours of a laboratory class per week (or 26 contact hours in total) per semester will be considered as one credit. A laboratory course may have half integer credits with a minimum of 1 credit.

3.3.3 Seminar, Thesis, Projects, Monographs, Fieldwork, Viva etc.:

Will be assigned by the respective discipline.

3.4 Classification of the Courses:
The Bachelor's degree courses will be classified into several groups and the syllabus committee will finalize the curricula selecting courses from the groups shown below.
3.4.1 Major Courses:
A student has to take at least 70% courses from his/her own discipline. Out of these courses a section will be identified as core courses and every student of a particular discipline will be required to take those courses.

3.4.2 Non-Major Courses:
Every student is required to take at least 20% (including mandatory) courses from related disciplines. If any Non-Major course is declared as Core course a student is required to take that course to graduate. The Non-Major courses will be designed, offered and graded by the offering disciplines.

3.4.3 Other Courses:
After completion of the required mandatory, major and non-major courses a student may take few other courses of his/her choice not directly related to his/her discipline to fulfill the total credit requirement.

3.4.4 Credit-Only Courses:
The credit of these Credit-Only courses will be added to the total credits if passed but will not affect the CGPA as there will be no grades for these courses.

4. Course Registration

4.1 Registration:
A student has to register for his/her courses and pay necessary dues within the first two weeks of every semester. Departmental student advisor will advise every student about his/her courses and monitor his/her performances. A student at any level is expected to register the courses at his level provided he/she does not have any incomplete courses from previous levels. A student will not be allowed to appear in the examination if his/her semester and examination fee is not cleared.

4.2 Minimum and Maximum Credits:
A student, if s/he is not a clearing graduate, has to register for at least 12 credits minimum and 30 credits maximum every semester.

4.3 Incomplete Courses:
(i) If a student has incomplete courses, he/she has to register his/her available incomplete courses from preceding levels before s/he can register courses from current or successive levels. If an incomplete course is not offered in a given semester the student has to take the courses when it is offered next time. A student with incomplete courses will not be eligible for Distinction.
(ii) A student to register his/her incomplete courses, if offered, from proceeding semesters before s/he can register courses from current or successive semester, otherwise s/he takes the courses when the desired course is offered next time. A

student will not be allowed to take 100 and 300 level and 200 and 400 level courses simultaneously. 100 level courses mean courses of 1st and 2nd semesters, 200 level courses mean courses of 3rd and 4th semesters and so on.

4.4 Course Withdrawal:
A student can withdraw a course by a written application to the Controller of Examinations through the Head of the discipline on or before the last day of instruction. The Controller of Examinations will send the revised registration list to the disciplines before the examination. There will be no record of the course in transcript if the course is withdrawn.

4.5 Course Repetition:
If a student has to repeat a failed or incomplete course and that course is not offered any more, the discipline may allow him/her to take an equivalent course from the current syllabus. For clearing graduates if any incomplete course is not offered in the running semester, the discipline may suggest a suitable course to complete the credit requirement.

5. Graduation Criteria

5.1 Major Degree:

5.1.1 Total Credits:
School of Physical Sciences, School of Social Sciences and School of Management and Business Administration have a requirement of 140 credits to graduate from its disciplines. School of Applied Sciences and Technology, School of Life Sciences and School of Agriculture and Mineral Science have requirement of 160 (200 for Architecture) credits for graduation.

5.1.2 Total Years:
A regular student is expected to graduate in 8 semesters (4 years) or in 10 semesters (5 years) for the discipline of Architecture. A student may graduate in shorter time period if s/he is willing to take extra courses in a systematic way. A student will be given 4 (2 years) extra semesters in addition to 8/10 semesters to complete his/her degree. The regular examination year will be identified by the session and the end-month (June or December) of the semester the student graduates.

5.1.3 Early Graduation:
A student may graduate early by completing courses in advance, in that case he does not need to pay tuition or get admission in subsequent semesters. However a student will not be able to start master's degree one session earlier unless he graduates two semesters early.

5.1.4 Minimum Credit for a Clearing Graduate:
For a clearing graduate (8th and subsequent semesters) condition for maximum and minimum credit requirements is relaxed.

5.1.5 Break in study:
Those students who have not been able to achieve their degrees by participating in the ascertained 12th (for ARC department 14th) semester final exams will have the opportunity to do so by enrolling into 2 (two) running semesters back to back if after the publications of their results of the 12th (for ARC department 14th) semester final exam, it becomes evident that they have completed at least 80% of their total credits. In case of such students, on the tabulation sheet, result sheet, certificate, transcript, grade sheet, etc., number of total semesters shall be stated instead of the word "Irregular." As for irregular students, studentship shall be annulled after the aforesaid 2 (two) semesters have come to an end.

5.2 Second Major Degree:

5.2.1 Total Credits:
A student may apply for a second major degree if he/she completes an extra 28-36 credit requirement designated by the offering discipline.

5.2.2 Total Semesters:
A student has to complete the credit requirement of second major degree within 8 regular and 4 extra semesters.

5.2.3 Requirement of Major Degree:
A student will not be given a second major degree if he/she fails to complete his regular major degree. A student will not be allowed to enroll in Masters program before completion of his/her second major degree even if he/she complete his/her major degree requirement.

5.2.4 Registration Criteria:
An offering discipline will decide on the number of seats for second major, enrollment criteria and get it approved from the academic council. Students willing to get a second major have to apply to the offering discipline for enrollment and the discipline will enroll them as per the admission criteria. During registration enrolled students have to get their courses approved from the offering department completing a separate registration form.

5.2.5 Class Routine:
After enrollment a regular student may start taking the second major courses starting 3rd semester. The class routine may be arranged to accommodate the student need.

5.2.6 Certificate and Mark sheet:
A student completing the requirement will be given an additional certificate and grade sheet for his second major degree.

6. Examination System

A student will be evaluated continuously in the courses system, for theoretical classes s/he will be assessed by class participation, assignments, quizzes, mid-semester

examinations and final examination. For laboratory work s/he will be assessed by observation of the student at work, viva-voce during laboratory works, from his/her written reports and grades of examinations designed by the respective course teacher and the examination committee.

6.1 Distribution of Marks:

The marks of a given course will be as follows:

Class Attendance	10%
Quiz abd Assignments	10%
Mid-Semester Examinations	20%
Final Examination	60%

6.1.1 Class Participation:

The marks for class participation will be as follows:

Attendance (Percentage)	Marks	Attendance (Percentage)	Marks	Attendance (Percentage)	Marks
95 and above	10	80 to 84	7	65 to 69	4
90 to 94	9	75 to 79	6	60 to 64	3
85 to 89	8	70 to 74	5	Less than 60	0

A student will not be allowed to appear at the examination of a course if his/her class attendance in that course is less than 50%.

6.1.2 Assignments and Mid-Semester Examinations:

There should be at least two mid-semester examinations for every course. The course teacher may decide the relative marks distribution between the assignments, tutorial and mid-semester examinations, however at least 50% contribution should come from the mid-semester examinations. The answer script should be returned to the students as it is valuable to their learning process.

6.1.3 Final Examination:

The final examination will be conducted as per the Semester Examination Ordinance. (a) Duration of the Final Examination: There will be a 3-hour final examination for every course of 3 credits or more after the 13th week from the beginning of the semester. Courses less than 3 credits will have final examination of duration 2 hours. (b) Evaluation of Answer Script: The students of the School of Applied Science and Technology and the School of Agriculture and Mineral Sciences will have two answer scripts to answer separate questions during final examination. Two separate

examiner will grade the two scripts separately and the marks will be added together to get the final mark. For the students of the other schools there will be a single answer script which will be evaluated by two examiners. The two marks will be averaged and if the marks by the two examiners differ by 20% or more the concerned answer scripts will be examined by a third examiner and the two closest marks among the three will be averaged to get the final mark.

7. Grading System

7.1 Letter Grade and Grade Point:

Letter Grade and corresponding Grade-Point for a course will be awarded from the roundup marks of individual courses as follows:

Numerical Grade	Letter Grade	Grade Point
80% and above	A+	4.00
75% to less than 80%	A	3.75
70% to less than 75%	A-	3.50
65% to less than 70%	B+	3.25
60% to less than 65%	B	3.00
55% to less than 60%	B-	2.75
50% to less than 55%	C+	2.50
45% to less than 50%	C	2.25
40% to less than 45%	C-	2.00
Less than 40%	F	0.00

7.2 Calculation of Grades

7.2.1 GPA:

Grade Point Average (GPA) is the weighted average of the grade points obtained in all the courses completed by a student in a semester.

7.2.2 CGPA:

Cumulative Grade Point Average (CGPA) of only major and both major and second major degree will be calculated by the weighted average of every course of previous semesters along with the present semester. For clearing graduates if the roundup value of the third digit after decimal is nonzero the second digit will be incremented by one. A student will also receive a separate CGPA for his second major courses.

7.2.3 F Grades:
A student is given an 'F' grade if he fails or is absent in the final examination of a course. If a student obtains an 'F' grade his grade will not be counted for GPA and s/he has to repeat the course. An 'F' grade will be in his/her record and s/he will not be eligible for Distinction.

8. Distinction

8.1 Distinction:
Candidates for four-year Bachelor degree will be awarded the degree with Distinction if his/her overall CGPA is 3.75 or above. However, a student will not be considered for Distinction if (a) s/he is not a regular student (has semester drop, incomplete courses in any semester or break of study) (b) has 'F' grade in one or more courses.

Ref.: This Ordinance was approved in the 126th Academic Council (26 June 2013). Clause 3.4.1 was cancelled in 127th Academic Council (27 August 2013). 128th Academic Council (21 November 2013) decided to make it effective from 01 January 2014.

শাহজালাল বিজ্ঞান ও প্রযুক্তি বিশ্ববিদ্যালয়ের বিএনসিসি ক্যাডেটদের জন্য ঐচ্ছিক বিষয় হিসেবে নির্ধারিত

MSC004 (3.0 Credits)
MILITARY SCIENCE (সামরিক বিজ্ঞান)

পঠিত বিষয় (তত্ত্বীয় ও ব্যবহারিক): বি এন সি সি'র ইতিহাস-ঐতিহ্য, বি এন সি সি'র সাংগঠনিক কাঠামো, মহান স্বাধীনতা যুদ্ধের পঠভূমি ও কারণ, স্বাধীনতা যুদ্ধের সেক্টর সমূহ, ড্রিল, কুচকাওয়াজ, ম্যাপ রিডিং, যুদ্ধের নানা কৌশল, যুদ্ধে ব্যবহৃত অস্ত্রের পরিচয়, বাংলাদেশের সশস্ত্র বাহিনীর পরিচয়, নেতৃত্বের বৈশিষ্ট্য, শরীর চর্চা, প্রাথমিক চিকিৎসা, সমাজ সেবা, দুর্যোগ ব্যবস্থাপনা, ভুমিকম্প ব্যবস্থাপনা, ঘূর্ণিঝড় ব্যবস্থাপনা, অগ্নি নির্বাপনের কৌশল, সাংস্কৃতিক প্রশিক্ষণ ইত্যাদি।

সহায়ক গ্রন্থ :
বি এন সি সি: সামরিক বিজ্ঞান সদর দপ্তর কর্তৃক নির্ধারিত ও প্রকাশিত।

Department of Computer Science and Engineering
Shahjalal University of Science & Technology
Sylhet-3114, Bangladesh
Syllabus for B. Sc. (Engg.) Program
Session: 2020-21

Vision Statement

The Department of Computer Science and Engineering, SUST intends to provide an excellent educational environment in order to develop professionals with strong technical and research backgrounds

Mission

M1. To provide quality education in both theoretical and applied foundations of Computer Science and Engineering.

M2. Tocreate highly skilled computer engineers, capable of doing research and also develop solutions for the betterment of the nation.

M3. To inculcate professional and ethical values among students.

M4. To support society by participating in and encouraging technology transfer

Program Name: B.Sc. (Engg.) in Computer Science and Engineering

Program Educational Objectives (PEO)

Program Educational Objectives (PEOs) are broad statements that describe what graduates are expected to attain within a few years of graduation. Program educational objectives are based on the needs of the program's constituencies.

The entity has set the following PEOs for the B.Sc. (Engg.) program in Computer Science and Engineering major.

PEO1. To provide students with a strong foundation in the mathematical, scientific and engineering fundamentals necessary to formulate, solve and analyze engineering problems and to prepare them for graduate studies, R&D, consultancy and higher learning.

PEO2. To develop an ability to analyze the requirements of the software, understand the technical specifications, design and provide novel engineering solutions and efficient product designs.

PEO3. To provide exposure to emerging cutting edge technologies, adequate training opportunities to work as teams on multidisciplinary projects with effective communication skills and leadership qualities.

PEO4. To prepare the students for a successful career and work with values and social concerns bridging the digital divide and meeting the requirements of local and multinational companies.

PEO5. To promote student awareness on life-long learning and to introduce them to professional ethics and codes of professional practice.

PEO to Mission Statement Mapping

Mission/PEO	PEO1	PEO2	PEO3	PEO4	PEO5
M1	X	X	X		
M2	X	X	X	X	
M3				X	X
M4			X		X

Program Learning Outcome (PLO)

After graduation from this program in CSE, the graduates will be able to:

PLO1. Apply knowledge of science, technology, computing andengineering in different aspects of their lifelong activities.

PLO2. Analyze a problem, identify and define the computing requirements appropriate to its solution.

PLO3. Design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.

PLO4. Use current techniques, skills, and tools necessary for computing practice.

PLO5. Apply mathematical foundations, algorithmic principles and computer science theory in modeling systems demonstrating tradeoffs and complexities involved in a design choice.

PLO6. Function effectively on teams to accomplish a common goal and communicate effectively with a range of audiences.

PLO7. Understand professional, ethical, legal, security and social issues and responsibilities.

PLO8. Analyze the local and global impact of computing on individuals, organizations and society.

PLO9. Engage in lifelong learning and grow capabilities of critical thinking and research

Program Objectives (PEO/PO) to Program Learning Outcome (PLO) Mapping

PLO/PEO	PEO1	PEO2	PEO3	PEO4	PEO5
PLO 1	X	X	X		
PLO 2	X	X			
PLO 3	X	X		X	
PLO 4			X	X	X
PLO 5	X	X			
PLO 6			X	X	
PLO 7				X	X
PLO 8				X	
PLO 9	X				X

Graduate Profile:

Graduate profiles are descriptions of attributes, or knowledge, skills and attitudes, which a university community intends its graduates will develop through their study

to equip them for their future education or employment. Students graduating from the department of CSE, SUST should have gained the following attributes.

a. Intellectual skills in Science and Engineering
b. Practical and problem solving skills
c. Numeracy and analytical skills
d. Entrepreneurship and innovation skills
e. Communication skills
f. Interpersonal, teamwork and leadership skill
g. Self-management & personal development skills
h. Commitment to community, country and humanity

Semester wise Curriculum Breakdown:

One-semester credit hour represents one class hour or two laboratory hours per week. An academic semester represents13 weeks of classes exclusive to final exams. Semester wise breakdown of the curriculum structure for 2020-21 session are shown.

Department of Computer Science and Engineering
Undergraduate Program
Session 2020-2021

First Year: 1st Semester

Course No.	Course Title	Hours/Week		Credits	Prerequisite
		Theory	Lab		
CSE 133	Structured Programming Language	3	0	3.0	
CSE 134	Structured Programming Language Lab	0	6	3.0	
CSE 143	Discrete Mathematics	3	0	3.0	
EEE 109D	Electrical Circuits	3	0	3.0	
EEE 110D	Electrical Circuits Lab	0	3	1.5	
MAT 101D	Coordinate Geometry And Linear Algebra	3	0	3.0	
ENG 101D	Effective Communication in English	2	0	2.0	
ENG 102D	English Language Lab 1	0	2	1.0	
SSS 100D	History of the Emergence of Independent Bangladesh	3	0	3.0	
Total		**17**	**11**	**22.5**	

First Year: 2nd Semester

Course No.	Course Title	Hours/Week		Credits	Prerequisite
		Theory	Lab		
CSE 137	Data Structure	3	0	3.0	CSE 133

CSE 138	Data Structure Lab	0	4	2.0	
CSE 147	Engineering Ethics and Cyber Law	2	0	2.0	
CSE 150	Project Work I	0	2	1.0	
EEE 111D	Electronic Devices and Circuits	3	0	3.0	EEE 109D
EEE 112D	Electronic Devices and Circuits Lab	0	3	1.5	
IPE 106D	Engineering Graphics	0	3	1.5	
IPE 108D	Workshop Practice	0	2	1.0	
PHY 103D	Mechanics, Wave, Heat & Thermodynamics	3	0	3.0	
MAT 103D	Calculus	3	0	3.0	
Total		**14**	**14**	**21.0**	

Second Year: 1st Semester

Course No.	Course Title	Hours/Week		Credits	Prerequisite
		Theory	Lab		
CSE 233	Object Oriented Programming Language	3	0	3.0	CSE 133
CSE 234	Object Oriented Programming Language Lab	0	4	2.0	
CSE 237	Algorithm Design & Analysis	3	0	3.0	CSE 137
CSE 238	Algorithm Design & Analysis Lab	0	4	2.0	
BUS 203	Cost and Management Accounting	3	0	3.0	
PHY 202D	Basic Physics Lab	0	3	1.5	
STA 202D	Basic Statistics & Probability	3	0	3.0	
Total		**12**	**11**	**17.5**	

Second Year: 2nd Semester

Course No.	Course Title	Hours/Week		Credits	Prerequisite
		Theory	Lab		
CSE 239	Numerical Analysis	2	0	2.0	
CSE 240	Numerical Analysis Lab	0	3	1.5	
CSE 241	Introduction to Data Science	2	0	2.0	
CSE 242	Introduction to Data Science Lab	0	3	1.5	
CSE 247	Theory of Computation	3	0	3.0	
CSE 252	Introduction to Competitive Programming	0	3	1.5	
EEE 201D	Digital Logic Design	3	0	3.0	EEE 109D, EEE 111D
EEE 202D	Digital Logic Design Lab	0	4	2.0	
ECO105D	Principles of Economics	3	0	3.0	
MAT204D	Complex Variables, Laplace Transform and Fourier Series	3	0	3.0	
CSE 250	Project Work II	0	2	1.0	
Total		**16**	**15**	**23.5**	

Third Year: 1st Semester

Course No.	Course Title	Hours/Week		Credits	Prerequisite
		Theory	Lab		
CSE 333	Database System	3	0	3.0	
CSE 334	Database System Lab	0	4	2.0	
CSE 335	Operating System and System Programming	3	0	3.0	
CSE 336	Operating System and System Programming Lab	0	3	1.5	
CSE 341	Software Engineering & Design Patterns	3	0	3.0	
CSE 342	Software Engineering & Design Patterns Lab	0	3	1.5	
CSE 344	Web Technologies	0	4	2.0	
CSE 365	Communication Engineering	2	0	2.0	
CSE 366	Communication Engineering Lab	0	2	1.0	
CSE 367	Microprocessor and Interfacing	3	0	3.0	EEE 201D
CSE 368	Microprocessor and Interfacing Lab	0	3	1.5	
Total		**14**	**19**	**23.5**	

Third Year: 2nd Semester

Course No.	Course Title	Hours/Week		Credits	Prerequisite
		Theory	Lab		
CSE 325	Digital Signal Processing	3	0	3.0	MAT 103D
					MAT 204D
CSE 326	Digital Signal Processing Lab	0	3	1.5	
CSE 329	Computer Architecture	3	0	3.0	
CSE 337	Artificial Intelligence	3	0	3.0	
CSE 338	Artificial Intelligence Lab	0	3	1.5	
CSE 361	Computer Networking	3	0	3.0	CSE 365
CSE 362	Computer Networking Lab	0	3	1.5	
CSE 376	Technical Writing And Presentation	0	4	2.0	
CSE 350	Project Work III	0	4	2.0	
Total		**12**	**17**	**20.5**	

Fourth Year: 1st Semester

Course No.	Course Title	Hours/Week		Credits	Prerequisite
		Theory	Lab		
CSE 421	Software Project Management	2	0	2.0	
CSE 461	Introduction to Computer Security	3	0	3.0	
CSE 462	Introduction to Computer Security Lab	0	3	1.5	
CSE 475	Machine Learning	3	0	3.0	

CSE 476	Machine Learning Lab	0	3	1.5	
CSE 4**	Option I	3	0	3.0	
CSE 4**	Option I Lab	0	3	1.5	
CSE 4**	Thesis / Project	0	4	2.0	
Total		**11**	**13**	**17.5**	
CSE 450	Project	0	4	2.0	
CSE 480	Thesis	0	4	2.0	

Fourth Year: 2nd Semester

Course No.	Course Title	Hours/Week		Credits	Prerequisite
		Theory	Lab		
CSE 493	Computer Graphics	3	0	3.0	
CSE 494	Computer Graphics Lab	0	3	1.5	
CSE 4**	Option II	3	0	3.0	
CSE 4**	Option II Lab	0	3	1.5	
CSE 4**	Thesis / Project	0	8	4.0	
CSE 484	Viva Voce	0	2	1.0	
Total		**6**	**16**	**14.0**	
CSE 452	Project	0	8	4.0	
CSE 482	Thesis	0	8	4.0	

List of Optional Courses

	Course Title	Hours/Week	Credits	

Course No.		Theory	Lab		Prerequisite
CSE439	Compiler Construction	3	0	3.0	
CSE440	Compiler Construction Lab	0	3	1.5	
CSE453	Cloud Computing	3	0	3.0	
CSE454	Cloud Computing Lab	0	3	1.5	
CSE455	Advanced Database System	3	0	3.0	
CSE456	Advanced Database System Lab	0	3	1.5	
CSE457	Mobile and Wireless Communication	3	0	3.0	CSE 365
CSE458	Mobile and Wireless Communication Lab	0	3	1.5	
CSE459	Advanced Data Structure and Algorithm	3	0	3.0	CSE 137, CSE 237
CSE460	Advanced Data Structure and Algorithm Lab	0	3	1.5	
CSE463	Security Engineering	3	0	3.0	
CSE464	Security Engineering Lab	0	3	1.5	
CSE465	Internet of Things	3	0	3.0	
CSE466	Internet of Things Lab	0	3	1.5	
CSE467	VLSI Design	3	0	3.0	EEE 201
CSE468	VLSI Design Lab	0	3	1.5	
CSE469	Bio-informatics	3	0	3.0	CSE 237
CSE470	Bio-informatics Lab	0	3	1.5	
CSE471	Digital Image Processing	3	0	3.0	

CSE472	Digital Image Processing Lab	0	3	1.5	
CSE473	Natural Language Processing	3	0	3.0	CSE 247
CSE474	Natural Language Processing Lab	0	3	1.5	
CSE477	Contemporary Course on Computer Science & Engineering I	3	0	3.0	
CSE478	Lab on Contemporary Course on Computer Science & Engineering I	0	3	1.5	
CSE487	Contemporary Course on Computer Science & Engineering II	3	0	3.0	
CSE488	Lab on Contemporary Course on Computer Science & Engineering II	0	3	1.5	
CSE485	Distributed and Parallel Computing	3	0	3.0	
CSE486	Distributed and Parallel Computing Lab	0	3	1.5	
CSE495	Human Computer Interaction	3	0	3.0	
CSE496	Human Computer Interaction Lab	0	3	1.5	
CSE497	Neural Network and Deep Learning	3	0	3.0	STA 202, MAT 103D, CSE 475 and CSE 476
CSE498	Neural Network and Deep Learning Lab	0	3	1.5	

Condition to receive degree from the Department of CSE

			Remarks
1	Total Credit to complete	160	
2	Major courses (from CSE)	Non-core courses	CSE 325 CSE 326 Option I (Theory and Lab) Option II (Theory and Lab)
		Core courses	All major courses except mentioned above.
3	Non-Major courses (from related discipline)	Non-core courses	IPE 108D
		Core courses	All non-major courses except mentioned above
4	Other courses (not directly related to CSE discipline)		Any courses offered by SUST

***Core courses mean mandatory courses for the degree**

COURSE PROFILE
Shahjalal University of Science and Technology
School of Applied Sciences and Technology
Department of Computer Science and Engineering

First Year

Course Title: Structured Programming Language	**Credits:** 3.0
Course No.: CSE 133	**Contact hours:** 3 hours/week

Rationale:
To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow the students to write their own programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:
- To facilitate students with necessary knowledge about basic understanding of computer hardware and how a computer works.
- To make the students understand the basic terminology used in computer programming
- Helping the students to develop ability in how to write, compile and debug programs in C language
- Helping the students to develop ability in writing programs involving decision structures, loops, functions and pointers
- To make the students understand the basic data structures and their implementation
- To enhancing the skill on implementing different searching and sorting techniques
- To enhancing the skill on building up their own logics and implementing them while solving real-world problems

Course Contents:
Programming Language: Basic concept, Overview of programming languages, Problem Solving Techniques and Data Flow Diagram. **C-Language:** Preliminaries, Program constructs, variables and data types in C. Input and output. Character and formatted I/O; Arithmetic Expressions and Assignment statements; Control statement, Loops and Nested loops; break, continue, Goto, Decision making; Arrays, Functions; Arguments and local variables, Calling Functions and arrays. Recursion and Recursive functions; Structures within structure. Automatic, external, static variable, Files; File functions for sequential and Random I/O. Pointers; Pointers and structures, union; Pointer and functions; Pointer and arrays; Operation and Pointer; Pointer and memory addresses; Operations on Bits; Bit Operation; Bit field; Advanced features; Preprocessor and Macros, enumeration, Standard library. **Recursion:** Basic idea of recursion (3 laws-base case, call itself, move towards base case by state change), tracing output of a recursive function, applications: factorial, Fibonacci, tower of Hanoi, merge sort, permutation, combination. **Sorting:** Insertion sort, selection sort, bubble sort, merge sort, quick sort, distribution sort (counting sort, radix sort, bucket sort). **Searching:** Linear search, binary Search, application of Binary Search- finding element in a sorted array, finding nth root of a real number, solving equations. **Stack and Queue:** Basic stack operations (push/pop/peek), stack-class implementation using Array and linked list, in-fix to postfix expressions conversion and evaluation, balancing parentheses using stack, basic queue operations (enqueue, dequeue), circular queue/ dequeue, queue-class implementation using array and linked list, application- Josephus problem, palindrome checker using stack and queue.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand the concepts of computer hardware and how it works.
CLO 2	Recall the basic terminology used in computer programming
CLO 3	Construct, compile and debug programs in C language
CLO 4	Apply control-flow tools such as loop, if-else, etc.
CLO 5	Understand the usage of pointers
CLO 6	Understand basic data structures, their implementation, and application
CLO 7	Apply different searching and sorting techniques
CLO 8	Evaluate real-life problems using programming terminologies

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X		X						
CLO2	X	X	X						
CLO3		X	X	X					
CLO4		X	X	X					
CLO5		X	X	X					

CLO6		X	X	X					
CLO7		X	X	X					
CLO8	X	X							

Textbook

1. Schaum's Outline of Programming with C by Byron S. Gottfried
2. C: The Complete Reference by Herbert Schildt

Course Title: Structured Programming Language Lab	**Credits:** 3.0
Course No.: CSE 134	**Contact hours:** 6 hours/week

Rationale:

To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow students to write their own programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:

- To facilitate students with necessary knowledge about basic understanding of computer hardware and how a computer works.
- To make the students understand the basic terminology used in computer programming
- Helping the students to develop ability in how to write, compile and debug programs in C language
- Helping the students to develop ability in writing programs involving decision structures, loops, functions and pointers
- To make the students understand the basic data structures and their implementation
- To enhancing the skill on implementing different searching and sorting techniques
- To enhancing the skill on building up their own logics and implementing them while solving real-world problems

Course Contents:

Programming Language: Basic concept, Overview of programming languages, Problem Solving Techniques and Data Flow-Diagram. **Language:** Preliminaries,

Program constructs, variables and data types in C. Input and output. Character and formatted I/O; Arithmetic Expressions and Assignment statements; Control statement, Loops and Nested loops; break, continue, goto, Decision making; Arrays, Functions; Arguments and local variables, Calling Functions and arrays. Recursion and Recursive functions; Structures within structure. Automatic, external, static variable, Files; File functions for sequential and Random I/O. Pointers; Pointers and structures, union; Pointer and functions; Pointer and arrays; Operation and Pointer; Pointer and memory addresses; Operations on Bits; Bit Operation; Bit field; Advanced features; Preprocessor and Macros, enumeration, Standard library. **Recursion:** Basic idea of recursion (3 laws-base case, call itself, move towards base case by state change), tracing output of a recursive function, applications: factorial, Fibonacci, tower of Hanoi, merge sort, permutation, combination. **Sorting:** Insertion sort, selection sort, bubble sort, merge sort, quick sort, distribution sort (counting sort, radix sort, bucket sort). **Searching:** Linear search, binary Search, application of Binary Search- finding element in a sorted array, finding nth root of a real number, solving equations. **Stack and Queue:** Basic stack operations (push/pop/peek), stack-class implementation using Array and linked list, in-fix to postfix expressions conversion and evaluation, balancing parentheses using stack, basic queue operations (enqueue, dequeue), circular queue/ dequeue, queue-class implementation using array and linked list, application- Josephus problem, palindrome checker using stack and queue.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand the concepts of computer hardware and how it works.
CLO 2	Recall the basic terminology used in computer programming
CLO 3	Construct, compile and debug programs in C language
CLO 4	Apply control-flow tools such as loop, if-else, etc.
CLO 5	Understand the usage of pointers
CLO 6	Understand basic data structures, their implementation, and application
CLO 7	Apply different searching and sorting techniques
CLO 8	Evaluate real-life problems using programming terminologies

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9

CLO1	X		X			X			
CLO2	X	X	X			X			
CLO3		X	X	X		X			
CLO4		X	X	X		X			
CLO5		X	X	X		X			
CLO6		X	X	X		X			
CLO7		X	X	X		X			
CLO8	X	X				X			

Textbook

1. Schaum's Outline of Programming with C by Byron S. Gottfried

C: The Complete Reference by Herbert Schildt

Course Title: Discrete Mathematics	**Credits:** 3.0
Course No.: CSE 143	**Contact hours:** 3 hours/week

Rationale:

CSE students need to have a very strong logical and mathematical background and a course of discrete math is essential for that. This course also works as the mathematical foundation for future courses like data structure, algorithm, digital electronics and theory of computation.

Objectives:

- Help them conceptualize basic theories in mathematical reasoning and appreciate the precision of language and rigor required for mathematics.
- Help them conceptualize basic theories in combinatorial analysis to be able to solve counting problems.
- To facilitate necessary knowledge about how to work with discrete data structures like graphs and trees.
- To facilitate necessary knowledge about algorithmic techniques and to be able to implement in computer programs.
- Apply the knowledge of discrete mathematics in real life problems using modeling.

Course Contents:

Set, Relations, Functions: Set, Function, Representing Relations, Equivalence Relations.

Propositional Calculus: Propositions, Predicate and Quantifier.

Algorithms: Complexity, Divisions, Algorithm, Application of Number Theory.

Recursion: Sequences and summations, Recursive Definition and algorithm.

Combinatorial Analysis: Permutation and Combination, Divide and Conquer Algorithms, Generating Functions.

Graphs: Representation, Isomorphism, Connectivity, Euler and Hamilton path, Shortest path, Planer, Coloring.

Trees: Spanning trees, Rooted Trees, Binary Trees, Huffman Trees.

Boolean Algebra: Number System, Boolean Function, representing Boolean Function, Logic gate, Minimization of Circuits.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Memorize the basic concepts of sets, permutations, relations, graphs, trees
CLO 2	Represent discrete objects and relationships using abstract mathematical structures
CLO 3	Apply basic concepts of mathematical logic and proof
CLO 4	Employ mathematical reasoning in order to read, comprehend, and construct mathematical arguments
CLO 5	Evaluate whether an algorithm works well and perform analysis in terms of memory and time
CLO 6	Formulate and model problems with the concepts and techniques of discrete mathematics

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	**PLO 1**	**PL0 2**	**PLO 3**	**PLO 4**	**PLO 5**	**PLO 6**	**PLO 7**	**PLO 8**	**PLO 9**
CLO1	X	X			X				
CLO2	X	X			X				
CLO3	X	X			X				
CLO4	X	X			X				
CLO5	X	X			X				
CLO6	X	X			X				

Textbook

1. Discrete Mathematics and Its Applications by Kenneth H. Rosen

Course Title: Data Structure	**Credits:** 3.0
Course No.: CSE 137	**Contact hours:** 3 hours/week

Rationale:
To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:
- To explain the purpose and mathematical background of algorithm analysis
- To facilitate necessary knowledge about the abstract data types of stacks, queues and deques
- To familiarize with variety of ways that linearly and weakly ordered data can be stored, accessed, and manipulated
- To facilitate necessary knowledge about the characteristics and optimal behavior of hash tables for access and retrieval
- To provide the knowledge of various sorting algorithms and the run-time analysis required to determine their efficiencies
- To help them understand various tree traversal techniques and graph algorithms

Course Contents:
Internal Data Representation: Specification, representation, Asymptotic analysis: Recurrences, Substitution method and manipulation of basic data structures: arrays, records and pointers, linked lists, stacks, queues, recursion, trees, optimal search trees, heaps, disjoint sets. **Recursion:** permutation, combination. **Sorting:** merge sort, quick sort (randomized quick sort), distribution sort (counting sort, radix sort, bucket sort), lower bounds for sorting, external sort. **Binary Tree:** Binary tree representation using array and pointers, traversal of Binary Tree (in-order, pre-order and post-order). **Ternary tree, Binary Search Tree:** BST representation, basic operations on BST (creation, insertion, deletion, querying and traversing), application- searching, sets. Ternary search tree, Binary Index tree, Segment tree, RMQ (Range Minimum Query). **Searching: A**pplication of Binary Search- finding element in a sorted array, finding nth root of a real number, solving equations. **Heap:** Min-heap, max-heap, Fibonacci-heap, applications-priority queue, heap sort. **Set Operations& Disjoint Set:** Union find, path compression. **Huffman Coding Graph:** Graph representation (adjacency matrix/adjacency list), basic operations on graph (node/edge insertion and deletion), **Traversing a graph:**Review of Breadth first search (BFS), Depth first search (DFS), Topological Sort, Strongly Connected Components, Euler Path, Articulation Point, Bridge, Bi-connected Components, graph-bicoloring, Floodfill, Dijkstra's Shortest Path Algorithm, Bellman –Ford algorithm and negative cycle detection, Floyd-Warshall all pair shortest path algorithm, Johnson's algorithm, shortest path in Directed Acyclic Graph. **Minimum spanning tree:** Prim's algorithm and Kruskal's algorithm. **Self-Balancing Binary Search Tree:** AVL tree (rotation, insertion). **Set Operations:** Set representation using bitmask, set/clear bit, querying the status of a bit, toggling bit values, LSB, application of set operations. **String ADT:** The concatenation of two strings, the extraction of substrings, searching a string for a matching substring, parsing, Suffix tree, Suffix array.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Interpret the examples of relationships between data.
CLO 2	Analyze computational and memory complexities of algorithms that are used to manipulate data using standard data structures.
CLO 3	Apply stacks/queues/deques to store and extract sequential data.
CLO 4	Apply recursions to divide a problem and conquer the solution.
CLO 5	Decide when to use which data structure.
CLO 6	Differentiate between graphs and trees and make graphs/trees when these are best suited for manipulating the data.
CLO 7	Demonstrate efficiency in inserting data to a data structure and searching/retrieving data from a data structure.
CLO8	Design data structures to store and manipulate data while solving real life problems.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO1	X	X	X		X				X
CLO2	X	X	X		X				X
CLO3	X	X	X		X				X
CLO4	X	X	X		X				X
CLO5	X	X	X		X				X
CLO6	X	X	X		X				X
CLO7	X	X	X		X				X
CLO8	X	X	X		X				X

Textbook

1. Advanced Data Structures, Peter Brass
2. Data Structures – Seymour Lipschutz, Schaum's Outlines Series.
3. Introduction to Algorithms Thomas H. Cormen , Charles E. Leiserson

Course Title: Data Structure Lab	**Credits:** 2.0
Course No.: CSE 138	**Contact hours:** 4 hours/week

Rationale:

To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:

- To explain the purpose and mathematical background of algorithm analysis
- To facilitate necessary knowledge about the abstract data types of stacks, queues and deques
- To familiarize with variety of ways that linearly and weakly ordered data can be stored, accessed, and manipulated
- To facilitate necessary knowledge about the characteristics and optimal behavior of hash tables for access and retrieval
- To provide the knowledge of various sorting algorithms and the run-time analysis required to determine their efficiencies
- To help them understand various tree traversal techniques and graph algorithms

Course Contents:

Creation and Manipulation of linear data structures: linked list, stacks and queues. **Creation and Manipulation of non-linear data structures**: B-trees and heaps, disjoint sets. Implementing sorting, searching and hashing techniques, string processing.

Implement all the Contents related to the coursework CSE 137.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Interpret the examples of relationships between data.
CLO 2	Analyze computational and memory complexities of algorithms that are used to manipulate data using standard data structures.
CLO 3	Apply stacks/queues/deques to store and extract sequential data.
CLO 4	Apply recursions to divide a problem and conquer the solution.
CLO 5	Decide when to use which data structure.
CLO 6	Differentiate between graphs and trees and make graphs/trees when these are best suited for manipulating the data.
CLO 7	Demonstrate efficiency in inserting data to a data structure and searching/retrieving data from a data structure.
CLO8	Design data structures to store and manipulate data while solving real life problems.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO1	X	X	X			X			
CLO2	X	X	X			X			
CLO3	X	X	X			X			
CLO4	X	X	X			X			
CLO5	X	X	X			X			
CLO6	X	X	X			X			
CLO7	X	X	X			X			
CLO8	X	X	X			X			

Textbook

1. Advanced Data Structures, Peter Brass
2. Data Structures – Seymour Lipschutz, Schaum's Outlines Series.
3. Introduction to Algorithms Thomas H. Cormen , Charles E. Leiserson

Course Title: Engineering Ethics And Cyber Law	**Credits:** 2.0
Course No.: CSE 147	**Contact hours:** 2 hours/week

Rationale: This course consists of a sustained study of ethical and legal issues that arise in relation to employment in the public and private sectors, including allocation of resources, corporate and social responsibility, relationships, and discrimination. A main focus of this course will be on the ethical and legal standards governing

information technology. New technology creates ethical challenges for individuals around the globe, and applies to most persons regardless of whether they are employed in the information technology field or a more traditional occupation. The study of Cyber Ethics provides a framework for making ethical decisions that professionals are likely to encounter in the workplace. This course will not only focus on ethics but on the legal, economic, social, cultural and global impacts of decisions that are made in the context of professional occupations.

Objectives:

- To make students explore and understand ethics and boundaries of morality and technology.
- To help them understand, explore, and acquire a critical understanding of cyber law.
- To facilitate necessary knowledge about different rules for legal bindings.

Course Contents:

Ethics: Introduction. Meta Ethics: Objectivism and Relativism, Non-naturalism, Cognitivism and Non-Cognitivism, The epistemic problem for cognitivism, Moral relativism, Cross-cultural differences and similarities, Different Psychological Issues in Meta-ethics: Egoism and Altruism, Emotion and Reason, Male and Female morality. Normative Ethics: Goodness, Rightness, Consequentialism, Utilitarianism. Applied Ethics: Business Ethics, Environmental Ethics and Social Ethics, Computer and Information Ethics. Developing the ethical analysis skills and professional values.

Cyber Law: Module I: Introduction: Computers, Internet and their Impacts in Society; Need for Cyber Law in Social and International Perspectives; Overview of Cyber Law, Cyberspace; Building blocks of CyberSpace; Cyber Jurisprudence at International and National Level; Jurisdictional Aspects in Cyber Law. **Module II: Cyber Crimes & Legal Framework:** Cyber Crimes against Individuals, Institution and State; Hacking; Digital Forgery; Cyber Stalking/Harassment; Cyber Pornography; Identity Theft & Fraud; Cyber terrorism; Cyber Defamation; Different offences under ICT Act, 2006. **Module III: Intellectual Property Issues in CyberSpace:** Interface with Copyright Law; Interface with Patent Law; Trademarks & Domain Names Related issues. **Module IV: E Commerce:** Concept; E-commerce-Salient Features; Online approaches like B2B, B2C & C2C; Online contracts; Click Wrap Contracts; Applicability of Contract Act, 1872. **Module V:** Cyber Tribunal: Establishment of Cyber Tribunal, Trial Procedure of Cyber Tribunal, Bail Rules, Time Limit, Power of Investigation etc.; Cyber Appellate Tribunal: Establishment of Cyber Appellate Tribunal, Procedure and Power Cyber Appellate Tribunal, Appeal Procedure in case of not establishing Cyber Appellate Tribunal.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO1	Apply diverse viewpoints to ethical dilemmas in the information technology field and recommend appropriate actions
CLO2	Identify and analyze statutory, regulatory, constitutional, and organizational laws that affect the information technology professional
CLO3	Locate and apply case law and common law to current legal dilemmas in the technology field
CLO4	Distinguish enforceable contracts from non-enforceable contracts

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1							X	X	
CLO2							X	X	
CLO3							X	X	
CLO4							X	X	

Course Title: Project Work I	**Credits:** 1.0
Course No.: CSE 150	**Contact hours:** 2 hours/week

Rationale:

This is a project which enables the freshmen to apply their novel acquired knowledge to some of the basic real world problem solving.

Objectives:

- Helping the students to develop ability in real life problem solving
- To enhance skill on problem solving
- To help them apply the knowledge of programming, data structure and algorithm

Course Contents:

Any project based on C language including implementation of Data Structure is acceptable. Gaming project using the graphics.h library in C is preferable. Teachers must have to ensure every project is unique. Innovative project ideas should get extra weight to prevent imitating old projects.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO1	Apply programming knowledge to create visible products

CLO2	Enhance problem solving capability
CLO3	Outline and design logical platforms to divide a problem and solve it with scientific and technical knowledge
CLO4	Habituate to work as an efficient team member
CLO5	Present ideas and projects in front of audience

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO1		X	X	X		X			
CLO2		X	X	X		X			
CLO3		X	X	X		X			
CLO4		X	X	X		X			
CLO5		X	X	X		X			

Second Year

Course Title: Object Oriented Programming Language	**Credits:** 3.0
Course No.: CSE 233	**Contact hours:** 3 hours/week

Rationale:
Students wishing to build up their career in CSE need to develop software to solve problems and this course will help them learn the basics of OOP and OOP programming using JAVA.

Objectives:
- To help students conceptualize basic theories and principles of object-oriented programming;
- Helping the students to develop ability in applying the concepts of data encapsulation, inheritance and polymorphism to large-scale software
- To facilitate necessary knowledge about good programming practices and how to write modular codes with the help of OOP concepts.
- To provide the knowledge of packages, how to work with them. Also give students a training to code reusable programs with JAVA.
- To make students understand how to work with JAVA generic templates to design Classes and data structures that can work with different data types.

Course Contents:
Introduction to Java: History of Java**,** Java Class Libraries**,** Introduction to Java Programming**,** A simple Program**. Developing Java Application:** Introduction, Algorithms, Pseudo code, Control Structure, The If /Else Selection Structure, The While Repetition Structure, Assignment Operators, Increment and Decrement Operators, Primitive Data Types, Common Escape Sequence, Logical Operator. **Control Structure:** Introduction with the 'for' structure, the 'switch' structure, the 'do/while' structure, the 'break' and 'continue' structure. **Methods:** Introduction, Program Module in Java, Math Class Methods, Method Definitions, Java API Packages, Automatic Variables, Recursion, Method Overloading, Method of the Applet Class**. Arrays:** Introduction, Arrays, Declaring and Allocating Arrays, Passing Arrays to Methods, Sorting Arrays, Searching Arrays, Multiple-Subscripted Arrays. **Object-Based Programming:** Introduction, Implementing a Time Abstract Data Type with a Class, Class Scope, Controlling Access to Members, Utility Methods, Constructors, Using Overload Constructor, Using Set and Get Method, Software Reusability, Friendly Members, Finalizers, Static Class Members, Data Abstraction and Information Hiding. **Object-Oriented Programming:** Introduction, Super classes and Subclasses, Protected Members, Using Constructor and Finalizers in Subclasses, Composition vs. Inheritance, Introduction to polymorphism, Dynamic method building, Final Methods and Classes, Abstract Superclasses and Concrete Classes**. String and Characters, Graphics, Exception. Handling, Files and Stream, Java API, Utility Classes, 2D Graphics, GUI, Swing, Events.**

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand the principles of object-oriented programming
CLO 2	Understand how to apply OOP in real world large problems
CLO 3	Design modular codes with the help of OOP concepts
CLO 4	Identify how to integrate robustness, reusability, and portability into large-scale software development.
CLO 5	*Understand how to write reusable codes*
CLO 6	Understand how to effectively design Class and data structures

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO 9
CLO1	X		X	X					X
CLO2	X		X	X					X
CLO3	X		X	X					X
CLO4	X		X	X					X
CLO5	X		X	X					X
CLO6	X		X	X					X

Textbook

1. Introduction to Programming in Java**,** Robert Sedgewick & Kevin Wayne.
2. An Introduction to Object-Oriented Programming, Timothy Budd.
3. Java-How to Program by Deitel & Deitel.

Course Title: Object Oriented Programming LanguageLab	**Credits:** 3.0
Course No.: CSE 234	**Contact hours:** 6 hours/week

Rationale:

Students wishing to build up their career in CSE need to develop software to solve problems and this course will help them learn the basics of OOP and OOP programming using JAVA.

Objectives:

- To give students hands-on training to help them understand OOP concepts with the help of JAVA.
- Helping the students to develop ability in applying the concepts of data encapsulation, inheritance and polymorphism to large-scale software
- To facilitate necessary knowledge about good programming practices and how to write modular codes with the help of OOP concepts.
- To provide the knowledge of packages, how to work with them. Also give students a training to code reusable programs with JAVA.
- To make students understand how to work with JAVA generic templates to design Classes and data structures that can work with different data types.
- To enable students to debug their codes by giving them an in-depth idea about different syntax errors, exceptions, and how to fix them.
- To enable students to develop a usable project (software, game, etc) with the help of OOP concepts.

Course Contents:

Object-Oriented Programming: Classes and objects, Constructors and destructors, Encapsulation of class members and methods, manipulating objects. **Dynamic Memory Allocation:** Pointers to objects, Pointers and arrays, Call-by-reference and call-by-value. **Concept of Inheritance, Interface and Polymorphism:** Direct and indirect inheritance, Private and protected members of inherited class, Constructors and destructors under inheritance, Polymorphism, Abstract base classes. **Exceptions:** Error handling in program, Creating own exception. **Handing Files:** Input/Output streams, Processing files, Random access files. **Thread Programming:** Introduction to threads, Using threads to solve multi-tasking problems, Thread synchronization. **Client-Server programming:** Applet and Servlets, Introduction to JSP, Socket programming. GUI: Basic user interface design using Java swing. **Understanding Java Enterprise Level Works.**

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand OOP concepts and implement them with the help of JAVA
CLO 2	Understand how to apply OOP in real world large problems.
CLO 3	Implement programming practices and write modular codes with the help of OOP concepts.
CLO 4	Employ reusable codes.
CLO 5	Design Class and data structures effectively
CLO 6	Point out basic syntax errors and exceptions, and identify how to fix them
CLO 7	Develop a usable project (software, game, etc) with the help of OOP concepts.
CLO 8	Understand threading and work with them.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X		X	X		X			X
CLO2	X		X	X		X			X
CLO3	X		X	X		X			X

CLO4	X		X	X		X			X
CLO5	X		X	X		X			X
CLO6	X		X	X		X			X
CLO7	X		X	X		X			X
CLO8	X		X	X		X			X

Textbook

1. Introduction to Programming in Java**,** Robert Sedgewick & Kevin Wayne.
2. An Introduction to Object-Oriented Programming, Timothy Budd.
3. Java-How to Program by Deitel & Deitel.

Course Title: Algorithm Analysis and Design	**Credits:** 3.0
Course No.: CSE 237	**Contact hours:** 3 hours/week

Rationale:

To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:

- To familiarize with the asymptotic performance of algorithms
- To familiarize with rigorous correctness proofs for algorithms
- To demonstrate a familiarity with major algorithms and data structures
- To facilitate with necessary knowledge about important algorithmic design paradigms and methods of analysis
- To develop skills to synthesize efficient algorithms in common engineering design situations

Course Contents:

Analysis of Algorithm: Asymptotic analysis: Recurrences, Substitution method, Recurrence tree method, Master method **Hash Table:** Hash tables, hash function, open addressing, perfect hashing, single and multi probehasing. **Greedy Algorithms:** Elements and properties of Greedy algorithms, fractional knapsack, job scheduling with deadline. **Dynamic Programming:** Elements of DP (Optimal substructure, Overlapping sub problem), Coin change related problem, 0-1 knapsack, Longest Common Subsequence finding problem, LCS and LIS/LDS variations, Matrix Chain Multiplication.**Red black Tree and Binomial Heaps, Stassen's algorithm Network Flow:** Flow Networks, Max-Flow Min-cut theorem, Ford Fulkerson method and its limitation, Edmonds Karp algorithm, Maximum bipartite matching, minimum path cover, edge cover. **Backtracking/Branch-and-Bound:** Permutation, Combination, 8-queen problem, 15-puzzle problem, Graph Coloring, N-queen problem, Hamiltonian cycle, Branch and Bound in backtracking. For example in traveling salesman problems. **Geometric algorithm:** Segment-segment intersection, Convex-hull, Closest pair problem. **Number Theory:** Chinese Remainder Theorem, Euler phi, extended Euclid, application of prime factorization application of phi. **RSA public key generation,** NP Completeness, NP hard and NPcomplete problems. **String Matching Algorithms:** Naïve string matching algorithm, Rabin Karp algorithm, String matching with finite automata, Knuth Morris Pratt (KMP) algorithm, Trie, Suffix tree and Suffix Array. Basic combinatorics, Probability and Game theory. Least Common Ancestor, Range Minimum Query, Polynomials, DFT and FFT

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Analyze the computational and memory complexities of algorithms
CLO 2	Prove the correctness of algorithms
CLO 3	Explain how and why the algorithms work
CLO 4	Apply the algorithms to solve real life problems
CLO 5	*Decide when to use which algorithm*
CLO 6	Synthesize algorithms to design complex solutions
CLO 7	Explain why one algorithm works better than others in different scenarios

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO 3	PLO 4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X	X						X
CLO2	X	X	X						X
CLO3	X	X	X						X
CLO4	X	X	X						X
CLO5	X	X	X						X
CLO6	X	X	X						X
CLO7	X	X	X						X

Textbook
1. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson.
2. Algorithms by Robert Sedgewick and Kevin Wayne.

Course Title: Algorithm Analysis and Design Lab	**Credits:** 2.0
Course No.: CSE 238	**Contact hours:** 4 hours/week

Rationale:
To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:
- To familiarize with the asymptotic performance of algorithms.
- To familiarize with rigorous correctness proofs for algorithms.
- To demonstrate a familiarity with major algorithms and data structures.
- To facilitate with necessary knowledge about important algorithmic design paradigms and methods of analysis.
- To develop skills to synthesize efficient algorithms in common engineering design situations.

Course Contents:
Hash Table: Hash tables, hash function, open addressing, perfect hashing, single and multi probehasing. **Greedy Algorithms:** Elements and properties of Greedy algorithms, fractional knapsack, job scheduling with deadline. **Dynamic Programming:** Elements of DP (Optimal substructure, Overlapping sub problem), Coin change related problem, 0-1 knapsack, Longest Common Subsequence finding problem, LCS and LIS/LDS variations, Matrix Chain Multiplication. **Red black Tree and Binomial Heaps, Stassen's algorithm Network Flow:** Flow Networks, Max-Flow Min-cut theorem, Ford Fulkerson method and its limitation, Edmonds Karp algorithm, Maximum bipartite matching, minimum path cover, edge cover. **Backtracking/Branch-and-Bound:** Permutation, Combination, 8-queen problem, 15-puzzle problem, Graph Coloring, N-queen problem, Hamiltonian cycle, Branch and Bound in backtracking. For example in traveling salesman problems. **Geometric algorithm:** Segment-segment intersection, Convex-hull, Closest pair problem. **Number Theory:** Chinese Remainder Theorem, Euler phi, extended Euclid, application of prime factorization application of phi. **RSA public key generation,** NP Completeness, NP hard and NPcomplete problems.**String Matching Algorithms:** Naïve string matching algorithm, Rabin Karp algorithm, String matching with finite automata, Knuth Morris Pratt (KMP) algorithm, Trie, Suffix tree and Suffix Array. Basic combinatorics, Probability and Game theory. Least Common Ancestor, Range Minimum Query, Polynomials, DFT and FFT.
Implement all the Contents related to the coursework CSE 237.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Implement all the listed algorithms.
CLO 2	Construct bug free and efficient codes for the algorithms.
CLO 3	Construct efficient solution of complex problems using suitable algorithms
CLO 4	Analyze the computational and memory complexities of algorithms.
CLO 5	Prove the correctness of algorithms.
CLO 6	Explain how and why the algorithms work.
CLO 7	Apply the algorithms to solve real life problems.
CLO 8	Decide when to use which algorithm.
CLO 9	Synthesize algorithms to design complex solutions.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X	X	X					X
CLO2	X	X	X	X					X
CLO3	X	X	X	X					X
CLO4	X	X	X	X					X
CLO5	X	X	X	X					X
CLO6	X	X	X	X					X
CLO7	X	X	X	X					X
CLO8	X	X	X	X					X
CLO9	X	X	X	X					X

Textbook
1. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson.

2. Algorithms by Robert Sedgewick and Kevin Wayne.

Course Title: Numerical Analysis	**Credits:** 2.0
Course No.: CSE 239	**Contact hours:** 2 hours/week

Rationale:
There are many interesting or economically pressing problems that "closed form algebraic solutions" are not available. Numerical methods are the answer to that. This course helps us to know how fast errors cause problems and to find better algorithms that cause less error. Therefore, this course is indispensable for all students in almost all disciplines.

Objectives:
- To develop skills to derive appropriate numerical methods to solve algebraic and transcendental equations
- To facilitate with necessary knowledge about performing an error analysis for various numerical methods
- To provide basic knowledge of coding various numerical methods in a modern computer language like Matlab, Python

Course Contents:
Numerical analysis: Errors in numerical calculations. Error: Definitions, sources, examples. Propagation of Error. A general error formula. Taylor series and reminders. **Root finding:** The bisection method and the iteration method, the method of false position. Newton-raphson method. Roots of polynomials. **Methods of approximation theory:** Polynomial interpolation: Lagrange form, divided formula for interpolation. **Solution of systems of Linear equations:** Gaussian elimination. The pivoting strategy, Iteration method solution of tridiagonal systems. LU decomposition, matrix inverse. **Numerical solution of ordinary differential equations:** Euler's method (including modified form), Runge-Kutta method. **Numerical Integration:** Trapezoidal method. Simpson's method. Weddle's method; Eigenvalue problems for matrices, Use of computer to implement projects in numerical methods.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Perform an error analysis for a given numerical method by going through the stages (mathematical modeling, solving and implementation) of solving a particular physical problem
CLO 2	Demonstrate understanding of common numerical methods and how they are used to obtain approximate solutions to otherwise intractable mathematical problems.
CLO 3	Apply numerical methods to obtain approximate solutions to mathematical problems.
CLO 4	Derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations.
CLO 5	Analyze and evaluate the accuracy of common numerical methods.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO1	X	X	X		X				
CLO2	X	X	X		X				
CLO3	X	X	X		X				
CLO4	X	X	X		X				
CLO5	X	X	X		X				

Textbook
1. Numerical Methods for Engineers by Steven C. Chapra, Raymond P. Canale.
2. Introduction to Numerical Analysis by F.B. Hildebrand.

Course Title: Numerical Analysis Lab	**Credits:** 1.5
Course No.: CSE 240	**Contact hours:** 3 hours/week

Rationale:
This course introduces students to numerical methods for the solution of basic mathematical problems that cannot be solved by hand. The course aims to introduce students to the toolbox of widely-used numerical methods in computational science. Students will be able to apply these methods to problems in a variety of sciences.

Objectives:
- To familiarize with the numerical methods used in computational science
- To help to develop skills to apply numerical methods to problems in practice.
- To familiarize with, use, and understand software which uses numerical methods

- To facilitate with the knowledge about the role of numerical methods in science
- To provide basic knowledge of coding various numerical methods in a modern computer language like Matlab, Python

Course Contents:
The material presented in this course is intended to acquaint students with some of the elementary numerical methods found useful in the fields of computing and applied mathematics.

Tasks:
1. Utilize numerical techniques to find the roots of an equation.
2. Set up a difference table and use it to interpolate and extrapolate data, determine the algebraic equation which will approximate the data, and perform numerical differentiations.
3. Perform linear and non-linear regression analysis of a set of data points using the method of least squares.
4. Calculate definite integrals using numerical integration methods and comparing those methods.
5. Solve systems of equations using matrix computations on the computer.
6. Use number theory to develop a solution better than the sieve of Eratosthenes prime algorithm.
7. Solve Josephus problem and Tower of Hanoi problem.
8. Using summation factors to solve different recurrence problems.
9. Compute the probabilities of events using summation for some calculations of probabilities and averages.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Perform an error analysis for a given numerical method by going through the stages (mathematical modeling, solving and implementation) of solving a particular physical problem
CLO 2	Demonstrate understanding of common numerical methods and how they are used to obtain approximate solutions to otherwise intractable mathematical problems.
CLO 3	Apply numerical methods to obtain approximate solutions to mathematical problems.
CLO 4	Derive numerical methods for various mathematical operations and tasks, such as interpolation, differentiation, integration, the solution of linear and nonlinear equations, and the solution of differential equations.
CLO 5	*Analyze and evaluate the accuracy of common numerical methods.*
CLO 6	Implement numerical methods using contemporary technology
CLO 7	Construct efficient, well-documented code and present numerical results in an informative way.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO1	X	X	X	X	X				
CLO2	X	X	X	X	X				
CLO3	X	X	X	X	X				
CLO4	X	X	X	X	X				
CLO5	X	X	X	X	X				
CLO6	X	X	X	X	X				
CLO7	X	X	X	X	X				

Textbook
1. Numerical Methods for Engineers by Steven C. Chapra, Raymond P. Canale.
2. Introduction to Numerical Analysis by F.B. Hildebrand.

Course Title: Introduction to Data Science	**Credits:** 2.0
Course No.: CSE 241	**Contact hours:** 4 hours / week

Rationale:
Data Science is a rapidly evolving field that studies how to analyze and organize relevant data through appropriate data visualizations. The technical foundation of Data science arises from Mathematics, Statistics and Computer Science. Those with a technical background related to data science need an understanding of the data relevant to the particular problem application area. Those with expertise in the application area must acquire the relevant technical knowledge in order to effectively and accurately make use of data science tools and methodologies. This course will build the technical and analytical skills required to collect, clean, and model data and show a path to bring all of these skills together in the creation and presentation of a data analytics predictive model, software system, or visualization.

Objectives:
- To introduce the fundamentals of data analytics and data science.

- To facilitate knowledge about data visualizations and appropriate analysis.
- To acquaint students with the methods to store and access data from a variety of sources.
- To familiarize with techniques and tools for transformation of Data.
- To help to accumulate basic ideas about statistical methods, regression techniques, and machine learning algorithms to make sense out of data sets both large and small.

Course Contents:
Introduction to Data Science, The scope of Data Science, Descriptive Statistics and Exploratory Data Analysis. Data Scraping, Cleaning and Summarization. Statistical Significance and P-values. Principles of Visualizing Data. Building Models and Validating Models. Linear Algebra Review. Linear Regression and Logistic Regression. Large-scale Clustering. Mining Massive Datasets. Crowdsourcing and Ensemble Learning.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Explain Data Science and Big Data and their impact on real life situations
CLO 2	Outline procedures for collecting, cleaning and preparing data from necessary domain
CLO 3	Apply different methods to summarize and analyze data
CLO 4	Create visual representation of data, that has been abstracted in some schematic form, including attributes or variables for the units of information
CLO 5	Construct plans to solve nontrivial problems by combining different search and decomposition techniques
CLO 6	Design models to solve data dependent real life problems
CLO 7	Plan to collect data through crowdsourcing
CLO 8	Summarize information by mining massive datasets and extrapolate them

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO1	X	X			X			X	X
CLO2	X	X			X			X	X
CLO3	X	X			X			X	X
CLO4	X	X			X			X	X
CLO5	X	X			X			X	X
CLO6	X	X			X			X	X
CLO7	X	X			X			X	X
CLO8	X	X			X			X	X

Textbook
1. The Signal and the Noise: Why so many predictions fail but some don't, by Nate Silver, Penguin Press.
2. The Art of Data Science, by Roger D. Peng and Elizabeth Matsui.

Course Title: Introduction to Data Science Lab	**Credits:** 1.5
Course No.: CSE 242	**Contact hours:** 3 hours lab per week

Rationale:
Data Science is a rapidly evolving field that studies how to analyze and organize relevant data through appropriate data visualizations. The technical foundation of Data science arises from Mathematics, Statistics and Computer Science. Those with a technical background related to data science need an understanding of the data relevant to the particular problem application area. Those with expertise in the application area must acquire the relevant technical knowledge in order to effectively and accurately make use of data science tools and methodologies. This course will build the technical and analytical skills required to collect, clean, and model data and show a path to bring all of these skills together in the creation and presentation of a data analytics predictive model, software system, or visualization.

Objectives:
- To facilitate necessary knowledge about functionality of Data Science
- To demonstrate how Data Science solves different real-world problems
- To help solve different Data Science problems using appropriate Data visualization, organization and presentation
- To help to develop systems by assembling different solution techniques

Course Contents:
Introduction to Data Science, The scope of Data Science, Descriptive Statistics and Exploratory Data Analysis. Data Scraping, Cleaning and Summarization. Statistical

Significance and P-values. Principles of Visualizing Data. Building Models and Validating Models. Linear Algebra Review. Linear Regression and Logistic Regression. Large-scale Clustering. Mining Massive Datasets. Crowdsourcing and Ensemble Learning.
Python for data analysis. Data Wrangling with Python, including tools/libraries as Pandas, NumPy, and IPython. Machine Learning using Python.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Analyze large dataset.
CLO 2	Visualize large dataset
CLO 3	Organize large dataset
CLO 4	Solve problems like water-jug problem, the missionaries and cannibal problem
CLO 5	Implement different tools and libraries (Pandas, NumPy, and IPython etc.) provided by Python.
CLO 6	Implement different tools and libraries provided by Python for machine learning.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO1	X	X			X	X		X	X
CLO2	X	X			X	X		X	X
CLO3	X	X			X	X		X	X
CLO4	X	X			X	X		X	X
CLO5	X	X			X	X		X	X
CLO6	X	X			X	X		X	X

Textbook
1. The Signal and the Noise: Why so many predictions fail but some don't, by Nate Silver, Penguin Press.
2. The Art of Data Science, by Roger D. Peng and Elizabeth Matsui.

Course Title: Theory of Computation	**Credits:** 3.0
Course No.: CSE 247	**Contact hours:** 3 hours/week

Rationale:
Theory of computation lays a strong foundation for a lot of abstract areas of computer science. It is used in Artificial Intelligence, Natural Language Processing, Probability or Computer vision, in certain areas of mathematics like Number theory. So Students wishing to build up their career in such a field of CSE need to achieve a better understanding of this topic.

Objectives:
- To familiarize with the formalization of the notion of problems via formal languages
- To familiarize with the formalization of the notion of computation using "abstract computing devices" called automata
- To help them understanding a hierarchy of classes of problems or formal languages (regular, context-free, context-sensitive)
- To help them understanding a hierarchy of classes of automata (finite automata, pushdown automata, and Turing machines)
- To facilitate with the knowledge about complexity classes P and NP, and Intractability (NP-completeness)
- To facilitate with the knowledge about space complexity: NL-completeness and PSPACE-completeness

Course Contents:
Introduction to Theory of Computation.
Automata and Language Theory: Finite automata: Deterministic and nondeterministic finite automata and their equivalence, regular expressions, Closure properties, push-down automata, context free grammars, pumping lemmas and applications. **Context-free Grammars:** Definitions. Parse trees. The pumping lemma for CFLs and applications. Normal forms. General parsing. Sketch of equivalence with pushdown automata.
Computability Theory: Turing machines: Designing simple TMs. Variations in the basic model (multi-tape, multi-head, nondeterminism), Church-Turing thesis and evidence to support it through the study of other models, decidability, halting problem, reducibility, recursion theorem. **Complexity Theory:** Time and space measures, hierarchy theorems, complexity classes P, NP, L, NL, PSPACE, BPP and IP, complete problems, P versus NP conjecture, quantifiers and games, provably hard problems, relativized computation and oracles, probabilistic computation, interactive proof systems.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Describe the fundamental elements and branches of theory of computation and its features
CLO 2	Design grammar for a language
CLO 3	Design regular expression to generalize all the elements of a language
CLO 4	Design deterministic and nondeterministic state diagrams to find out acceptable and non-acceptable elements for a specific language
CLO 5	Analyze and evaluate grammars to prevent ambiguities in language components
CLO 6	Decide which problems are solvable and not solvable by a system
CLO 7	Apply transformation of grammar to meet different conditions
CLO 8	Analyze time complexity of an algorithm
CLO 9	Analyze space complexity of an algorithm

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO1	X	X			X				
CLO2	X	X			X				
CLO3	X	X			X				
CLO4	X	X			X				
CLO5	X	X			X				
CLO6	X	X			X				

Textbook

1. Introduction to the Theory of Computation by Michael Sipser, (Latest Edition).
2. Introduction to Languages and the Theory of Computation, by J. C. Martin.

Course Title: Project Work II	**Credits:** 1
Course No.: CSE 250	**Contact hours:** 2 hours/week

Rationale:

This course is based on project work. Target of this course is to involve students in real life software development which will help to increase their skill to reach the requirements of the software industry. Also this course will help students to improve their communication skill and to present their work in front of an audience.

Objectives:

- To facilitate necessary knowledge about latest technology
- To develop skills on software development
- To develop skills on teamwork and presentation

Course Contents:

Project focusing on an Object oriented programming approach and using standard algorithms is preferable. Every project should maintain a goal so that it can be used as a useful tool in the IT fields. Also innovative project ideas that require different types of scripting/programming languages or programming tools can be accepted with respect to the consent of the corresponding project supervisor.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Apply latest state of the art technologies
CLO 2	Design and implement ideas for complete software
CLO 3	Evaluate existing computer and mobile applications
CLO 4	Explain ideas to groups and present their noble findings

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO1	X		X	X	X	X			X
CLO2	X		X	X					X
CLO3		X	X	X					X

CLO4		X	X	X	X	X			X

Course Title: Introduction To Competitive Programming	**Credits:** 1.5
Course No.: CSE 252	**Contact hours:** 3 hours/week

Rationale:
This course is intended to facilitate students with advanced knowledge on advanced data structure and algorithms.

Objectives:
- To facilitate necessary knowledge about advanced data structures and algorithms
- To enhance the skill on problem solving
- To help to improve thinking process

Course Contents:
Data Structure: Trie Tree, BIT, Segment Tree, Splay Tree, MO's Algorithm, Square Root Decomposition, Heavy Light Decomposition, Persistent Data Structure (Segment Tree, Trie), DSU on Tree, Treap, K-D Tree, KNN Tree, Sparse Table. **String Processing:** KMP, Suffix Array, Suffix Automata, Suffix Tree, Palindromic Tree, Aho-Corasick, Manacher Algorithm, Extended KMP, Hashing (Rolling Hash). **Game Theory:** Nim Game, Sprague-Grundy Value, Green Hackenbush, Blue Red Hackenbush, Blue Red Green Hackenbush, Colon Principle, Fusion Principle. **Combinatorics & Probability:** Burnside Lemma, Inclusion Exclusion, Combination, Permutation, Catalan Number, Stirling Number, Probability, Expected Value. **Number Theory:** Chinese Remainder Theorem, Euler Phi, Extended Euclid, Prime Factorization, Mobius Function, Primitive Prime, Huge Mod. **Basic Math:** FFT, DFT, NTT, Gaussian Elimination, Matrix Exponentiation. **Basic Geometry:** Fundamental Concepts of Geometry, Closest Pair of Point, Convex Hull, Rectangle Union, Circle Union, Polygon Clipping, Line Sweep, Line Intersection.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Apply advanced data structures and algorithms
CLO 2	Improve problem solving skills
CLO 3	Design and interpret complex logics to solve complicated problems
CLO 4	Prepare themselves to deal with competitive environments by attending frequent problem solving contests

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL0 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO 1	X	X		X		X			X
CLO 2	X	X		X		X			X
CLO 3	X	X		X		X			X
CLO 4	X	X		X		X			X

Textbook
1. Competitive Programming 3 by Steven Halim.
2. 102 Combinatorial Problems by Titu Andreescu & Zuming Feng.
3. Problem-Solving Methods in Combinatoricsby Pablo Soberón.
4. Art of Programming Contest by Ahmed ShamsulArefin.
5. Programming Challenges: The Programming Contest Training Manual bySteven S Skiena, Miguel A. Revilla.

Third Year

Course Title: Database System	**Credits:** 3.0
Course No.: CSE 333	**Contact hours:** 3 hours/week

Rationale:
Database System course will concentrate on the principles, design, implementation and applications of database management systems.

Objectives:
- To facilitate knowledge about different issues involved in the design and implementation of a database system.
- To make students understand physical and logical database designs, database modeling, relational, hierarchical and network models
- To provide the knowledge of data manipulation language to query, update, and manage a database
- To help to develop an understanding of essential DBMS concepts such as: database security, integrity, concurrency, distributed database, Client/Server (Database Server), Data Warehousing.
- To help apply the knowledge of design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing and implementing a DBMS

.

Course Contents:

Introduction: Purpose of Database Systems, Data Abstraction, Data Models, Instances and Schemes, Data Independence, Data Definition Language, Data Manipulation Language, Database Manager, Database administrator, Database Users, Overall System Structure, Advantages and Disadvantage of a Database Systems. Data Mining and analysis, Database Architecture, History of Database Systems. **Relationship Entity-Model:** Entities and Entity Sets, Relationships and Relationship Sets, Attributes, Composite and Multivalued Attributes, Mapping Constraints, Keys, Entity-Relationship Diagram, Reducing of E-R Diagram to Tables, Specialization, Generalization, Attribute Inheritance, Aggregation, Alternative E-R Notations, Design of an E-R Database Scheme. **Relational Model:** Structure of Relational Database, Fundamental Relational Algebra Operations, the Tuple Relational Calculus, the Domain Relational Calculus, Modifying the Database. **Relational Commercial Language:** SQL, Basic structure of SQL Queries, Query-by-Example, Nested Sub queries, Complex queries, Integrity Constraints, Authorization, Dynamic SQL, Recursive Queries. **Relational Database Design:** Pitfalls in Relational Database Design, Functional Dependency Theory, Normalization using Functional Dependencies, Normalization using Multivalued Dependencies, Normalization using join Dependencies, Database Design Process. **File and System Structure:** Overall System Structure, Physical Storage Media, File Organization, RAID, Organization of Records into Blocks, Sequential Files, Mapping Relational Data to Files, Data Dictionary Storage, Buffer Management. **Indexing and Hashing:** Basic Concepts, Ordered Indices, B+ -Tree Index Files, B-Tree Index Files, Static and Dynamic Hash Function, Comparison of Indexing and Hashing, Index Definition in SQL, Multiple Key Access. **Concurrency Control:** Schedules, Testing for Serializability, Lock-Based Protocols, Timestamp-Based Protocols, Validation Techniques, Multiple Granularity, Multiversion Schemes, Insert and Delete Operations, Deadlock Handling. **Distributed Database:** Structure of Distributed Databases, Trade-off in Distributing the Database, Design of Distributed Database, Transparency and Autonomy, Distributed Query Processing, Recovery in Distributed Systems, Commit Protocols, Concurrency Control.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Describe the fundamentals of Database systems
CLO 2	Differentiate between different database models
CLO 3	Design a Relational Database
CLO 4	Create queries in relational databases
CLO 5	Use indexing for databases
CLO 6	Implement concurrency control mechanisms
CLO 7	Implement distributed databases

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X			X				X	
CLO2	X			X				X	
CLO3	X			X				X	
CLO4	X			X				X	
CLO5	X			X				X	
CLO6	X			X				X	
CLO7	X			X				X	

Textbook

1. Database System Concepts – Abraham Silberschratz, Henry K. Korth, S. Sudarshan (5th edition)
2. Fundamentals of Database Systems - Benjamin/Cummings, 1994
3. Database Principles, Programming, Performance - Morgan Kaufmann 1994
4. A First Course in Database Systems - Prentice Hall, 1997
5. Database Management Systems, McGraw Hill, 1996

Course Title: Database System Lab	**Credits:** 2.0
Course No.: CSE 334	**Contact hours:** 4 hours/week

Rationale:

Database System LAB course will concentrate on the design and implementation of a database system and applying SQL query.

Objectives:

- To help them understand the different issues involved in the design and implementation of a database system
- To facilitate with knowledge about physical and logical database designs, database modeling, relational, hierarchical and network models.
- To help them understand and use data manipulation language to query, update, and manage a database
- To help them develop skill to design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing and implementing a DBMS.

Course Contents:
Introduction: MySQL, Oracle, SQL, Data types, SQL / PLSQL, Oracle Software Installation, User Type, Creating User , Granting. **Basic Parts of Speech in SQL:** Creating Newspaper Table, Select Command (Where, order by), Creating View, Getting Text Information & Changing it, Concatenation, Cut & paste string (RPAD, LPAD, TRIM, LTRIM, RTRIM, LOWER, UPPER, INIT, LENGTH, SUBSTR, INSTR, SOUNDEX). **Playing The Numbers:** Addition, Subtraction, Multiplication, Division, NVL, ABS, Floor, MOD, Power, SQRT, EXR, LN, LOG, ROUND, AVG, MAX, MIN, COUNT, SUM, Distinct, SUBQUERY FOR MAX, MIN. **Grouping things together:** Group By, Having, Order By, Views Renaming Columns with Aliases. **When one query depends upon another:** Union, Intersect, Minus, Not in, Not Exists. **Changing Data:** INSERT, UPDATE, MERGE, DELETE, ROLLBACK, AUTOCOMMIT, COMMIT, SAVEPOINTS, MULTI TABLE INSERT, DELETE, UPDATE, MERGE. **Creating And Altering tables & views:** Altering table, Dropping table, Creating view, Creating a table from a table. **By What Authority:** Creating User, Granting User, Password Management. **An Introduction to PL/SQL**: Implement a few problems using PL/SQL (e.g. Prime Number, Factorial, Calculating Area of Circle, etc.). **An Introduction to Trigger and Procedure:** Implement few problems using Trigger and Procedures. **An Introduction to Indexing:** Implement indexing using a large database and observe the difference of Indexed and Non-Indexed databases.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Create a database and necessary tables
CLO 2	Apply functions to process numeric data
CLO 3	Use SQL query for grouping data
CLO 4	Apply DDL to insert, update and manage database schema
CLO 5	Apply PL/SQL and Trigger to solve problems
CLO6	Design E-R diagrams to present a database scheme

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X			X		X		X	
CLO2	X			X		X		X	
CLO3	X			X		X		X	
CLO4	X			X		X		X	
CLO5	X			X		X		X	
CLO6	X			X		X		X	

Textbook
1 Database System Concepts – Abraham Silberschratz, Henry K. Korth, S. Sudarshan (5th edition)
2 Fundamentals of Database Systems - Benjamin/Cummings, 1994
3 Database Principles, Programming, Performance - Morgan Kaufmann 1994
4 A First Course in Database Systems - Prentice Hall, 1997
5 Database Management Systems, McGraw Hill, 1996

Course Title: Operating System and System Programming	**Credits:** 3.0
Course No.: CSE 335	**Contact hours:** 3 hours/week

Rationale:
This course presents fundamental concepts related to the design and implementation of operating systems. Topics include basic operating system structure, process scheduling, process and thread synchronization and concurrency, memory management, file systems.

Objectives:
- To acquaint students with the role of the operating system as a high level interface to the hardware.
- To make them understand basic idea about the low level implementation of CPU dispatch.
- To make them understand basic idea about the low level implementation of memory management.
- To provide knowledge about the performance trade-offs inherent in OS implementation

Course Contents:
Introduction: Operating Systems Concept, Computer System Structures, Operating System Structures, Operating System operations, Protection and Security, Special-Purpose Systems. **Fundamentals of OS:** OS services and components, multitasking, multiprogramming, time sharing, buffering, spooling **Process Management:** Process Concept, Process Scheduling, Process State, Process Management, Interprocess Communication, interaction between processes and OS, Communication in Client-Server Systems, Threading, Multithreading, Process Synchronization.**Concurrency control:** Concurrency and race conditions, mutual exclusion requirements, semaphores, monitors, classical IPC problem and solutions, Dead locks -

characterization, detection, recovery, avoidance and prevention. **Memory Management**: Memory partitioning, Swapping, Paging, Segmentation, Virtual memory - Concepts, Overlays, Demand Paging, Performance of demand paging, Page replacement algorithm, Allocation algorithms. **Storage Management**: Principles of I/O hardware, Principles of I/O software, Secondary storage structure, Disk structure, Disk scheduling, Disk Management, Swap-space Management, Disk reliability, Stable storage implementation. **File Concept:** File support, Access methods, Allocation methods, Directory systems, File Protection, Free Space management.**Protection & Security**: Goals of protection, Domain of protection, Access matrix, Implementation of access matrix, Revocation of access rights, The security problem, Authentication, One-time passwords, Program threats, System threats, Threat monitoring, Encryption, Computer-security classification. **Distributed Systems:** Types of Distributed Operating System, Communication Protocols, Distributed File Systems,Naming and Transparency, Remote File Access, Stateful Versus Stateless Service, File Replication. **Case Studies:** Study of a representative Operating Systems. **System Programming:** Introduction to System Programming and Linux / Unix, Shell Programming, C Language forSystem Programming, Make and Make files, Process and Signals, Threads, Inter process Communications, X- Window Programming, Principle of single and multi-user operating systems.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Compare the functionality of different computing hardware structures and Operating System Structures.
CLO 2	Discuss issues of Process Management including process structure, synchronization, scheduling and communication.
CLO 3	Discuss memory management issues including pages, segmentation and virtual memory.
CLO 4	Explain the operation of various File Management Algorithms.
CLO 5	Discuss the issues related to I/O, security, distributed systems

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO1			X		X				
CLO2			X		X				
CLO3			X		X				
CLO4			X		X				
CLO5			X		X				

Textbook

1. Operating System Concepts by Silberschatz& Galvin Wiley 2000 (7th Edition)
2. Operating Systems by Achyut S. Godbole Tata Mc Graw Hill (2nd Edition)

Course Title: Operating System and System Programming Lab	**Credits:** 1.5
Course No.: CSE 336	**Contact hours:** 3 hours/week

Rationale:

The goal of this course is to have students understand and appreciate the principles in the design and implementation of some of the features on operating systems software.

Objectives:

- To facilitate idea about Inter-process communication
- To facilitate the knowledge of CPU scheduling for processes
- To facilitate the knowledge of Process Synchronization using Critical section
- To facilitate the knowledge of Multi -Threading and Thread Synchronization
- To acquaint students with the knowledge of network operating system tasks through simulation/implementation

Course Contents:

Thread programming: Creating thread and thread synchronization. **Process Programming:** The Process ID, Running a New Process, Terminating a Process, Waiting for Terminated Child Processes, Users and Groups, Sessions and Process Groups. **Concurrent Programming:** Using fork, exec for multi-task programs. **File Operations:** File sharing across processes, System lock table, Permission and file locking, Mapping Files into Memory, Synchronized, Synchronous, and Asynchronous Operations, I/O Schedulers and I/O Performance. **Communicating across processes:** Using different signals, Pipes, Message queue, Semaphore, Semaphore arithmetic and shared memory.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand some features on Linux
CLO 2	Understand Process to process communication

CLO 3	Explain concept on managing multiprocessing system.
CLO 4	Understand Process Synchronization
CLO 5	Understand Multi -Threading and Thread Synchronization
CLO 6	Study on Network Operating System

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO1			X		X				
CLO2			X		X				
CLO3			X		X				
CLO4			X		X				
CLO5			X		X				
CLO6			X		X				

Textbook

1. The 'C' Odyssey UNIX-The Open, Boundless C by Meeta Gandhi, Tilak Shetty, Rajiv Shah.
2. Beginning Linux Programming by Neil Matthew and Richard Stones
3. Linux System Programming by Robert Love

Course Title: Software Engineering and Design Patterns	**Credits:** 3.0
Course No.: CSE 341	**Contact hours:** 3 hours/week

Rationale:

Software Engineering is about the discipline needed to develop high quality software that can be understood, maintained and adapted over long periods of time by many different people. The course attempts to foster an understanding of software quality: what it is, and how to achieve it. This can be done through the use of a team project running throughout the course, in which teams trade software modules with one another. By attempting to understand, assess, and modify one another's programs, students will gain insight into the nature of software quality, and why an ability to program is not sufficient for the construction of high quality software.

Objectives:

- To give students an insight about common software engineering processes and well-known practices.
- To make students understand the impact of requirement engineering and the proper way to do that.
- To make students understand basic design principles and how those principles can be utilized to make more modular and scalable programs.
- To help students develop skills that will enable them to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain.
- To provide knowledge of basic software measurement concepts and how to allocate resources from the perspective of a software manager or team lead.
- To facilitate students with the knowledge how to properly test their software and modern software verification and validation practices.

Course Contents:

Introduction: Introduction to Software Engineering, Software Development Process and Various Life Cycle Models. **Requirement Analysis:** Communication Techniques, Analysis Principles, Software Prototyping, Requirement Specification. **Analysis Modeling:** Steps of system analysis, Feasibility study, Economic and technical analysis, System specification, the elements of analysis model, Data modeling, Functional modeling and information flow, Behavioral modeling, Mechanics of structured analysis, Data Dictionary. **Software Design:** Design principles, Design Concepts, effective modular design, design heuristics, Data Design, Architectural Design process, Transformation mapping, Transaction mapping, interface design, human-computer interface design, procedural design. **Software Testing:** Testing fundamentals, test case design, white-box testing, black-box testing, testing GUIs, Unit testing, Integration testing, validation testing, system testing, debugging. **Maintenance:** Major maintenance activities, estimating maintenance cost and productivity. **Technical Metrics for Software:** Software quality, Framework for technical metrics, metrics for analysis and design models, source code, testing and maintenance. **Software Architecture:** Pipe and Filter, Object Oriented, Event Based, Layered System, Data-centered repository, Process Control Architectures. **Software Project Management:** Cost estimation, risk analysis, project scheduling. **Design Patterns:** Introduction to design patterns. Different Patterns: Strategy, Observer, Factory, Singleton, Command, Adapter, Facade, Template Method, Iterator, Composite, State, Proxy, Compound Patterns. **Formal Methods:** Formal Methods in Software Engineering: its need and application, Formal specifications, Formal Verifications, Introduction to Z Language, Formal methods and testing.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Develop and insight about common software engineering processes and well-known practices
CLO 2	Understand the impact of requirement engineering and the proper way to do that
CLO 3	Understand basic design principles and how those principles can be utilized to write modular codes
CLO 4	Develop skills that will enable them to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain
CLO 5	Understand basic software measurement concepts and how to allocate resources from the perspective of a software manager or team lead
CLO 6	Understand modern software verification and validation practices
CLO 7	Identify internal software architecture of different existing software
CLO 8	Apply appropriate design pattern while developing a software

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X	X					X	X
CLO2	X	X	X					X	X
CLO3	X	X	X					X	X
CLO4	X	X	X					X	X
CLO5	X	X	X					X	X
CLO6	X	X	X					X	X

Textbook

1. Software Engineering: A Practitioner's Approach- Roger S. Pressman.
2. Head First Design Patterns, Eric & Elisabeth Freeman, O'REILLY.

Course Title: Software Engineering and Design Patterns Lab	**Credits:** 1.5
Course No.: CSE 342	**Contact hours:** 3 hours/week

Rationale:

Software Engineering is about the discipline needed to develop high quality software that can be understood, maintained and adapted over long periods of time by many different people. The course attempts to foster an understanding of software quality: what it is, and how to achieve it. This can be done through the use of a team project running throughout the course, in which teams trade software modules with one another. By attempting to understand, assess, and modify one another's programs, students will gain insight into the nature of software quality, and why an ability to program is not sufficient for the construction of high quality software.

Objectives:

- To provide students a hands-on training on basic design principles and how those principles can be utilized to make more modular and scalable programs.
- To familiarize students with basic software engineering diagrams like (class diagram, state diagram, use-case diagrams, etc.) and how these diagrams can be used to describe a software from different viewpoints.
- To help students develop the ability of significant teamwork and project based experience
- To help develop skills that will enable the students to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain

Course Contents:

Software Engineering lab work is solely designed to attain hands-on experience of architectural design, documentation and testing of software so that students can develop the software following the documents only. Also this lab includes Introduction to UML, Introduction to CASE Tools and Introduction to MVC Pattern. **Step1 (Requirement Engineering)**: Choose a company/institute/client for which software will be developed (make sure that they will provide required information whenever necessary). Follow the steps for eliciting requirements and generate a use-case diagram. Also analyze the sufficiency of the requirement engineering outcome for steps to follow. **Step 2 (Analysis model to Architectural and Component level design)**: Generate Activity diagram, Data flow diagram(DFD), Class diagram, State diagram, Sequence diagram and follow other relevant steps for creating complete architectural and component level design of the target software. **Step 3 (User Interface design, Design evaluation, Testing strategies and Testing Tactics)**: Perform the user interface design with the help of swim lane diagram. Carry out the design evaluation steps. Generate all test cases for complete checking of the software using black box, white box testing concept. **Step 4 Software testing and debugging. Step 5 (Managing Software Projects)**: Analyze the estimation and project schedule.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand basic design principles and how those principles can be utilized to write modular codes
CLO 2	Understand how different diagrams can be used to describe a software from different viewpoints
CLO 3	Develop the ability of significant teamwork and project based experience
CLO 4	Acquire skills that will enable the students to construct software of high quality
CLO 5	Analyze requirements of a real life software
CLO 6	Estimate timeline for a fully functional software development

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X	X			X		X	X
CLO2	X	X	X			X		X	X
CLO3	X	X	X			X		X	X
CLO4	X	X	X			X		X	X
CLO5	X	X	X			X		X	X
CLO6	X	X	X			X		X	X

Textbook

1. Head First Software Development, O'Relly

Course Title: Web Technologies	**Credits:** 2.0
Course No.: CSE 344	**Contact hours:** 4 hours/week

Rationale:

Websites and web applications behold a very large portion of the software industry. Technologies related to this sector are getting updated very frequently. To keep pace with this continuously upgrading world students must have a very good rudimentary level knowledge of web technologies. Objective of this course is to make students introduced to this sector and prepare them for the industry by providing them necessary hands-on tools.

Objectives:

- To facilitate with rudimentary level knowledge of web technologies
- To provide knowledge on development of web applications
- To make students understand server-side, front-end technologies
- To make students understand Model View Controller model
- To provide knowledge on plug-ins, APIs, extensions

Course Contents:

Concepts of Web Engineering, Requirements Engineering and Modeling Web Applications, Web Application Architectures, Technologies and Tools for Web Applications, Testing and Maintenance of Web Applications, Usability and Performance of Web Applications, Security of Web Applications, The Semantic Web, design methods and technologies, interface design, usability of web applications, accessibility, testing, metrics, operation and maintenance of Web applications, security, and project management, client-side (XHTML, JavaScript, and CSS) and server-side (Perl and PHP) architecture, Web engineering concepts behind the frameworks of Joomla, Drupal, Wordpress. **Server-side technology:** LAMP, Web application frameworks (example: Silverlight, Adobe Flex), Web 2.0 and Web APIs. **Front-end technology:** HTML, XHTML, XML. CSS styling, layout, selector, Document object model and JavaScript. **Client-Programming:** Web APIs with JavaScript (example: Google Ajax API). **MVC:** Understanding Model, view and controller Model. **Understanding Web APIs:** REST, XML, JSON, RSS Parsing. **JavaScript Exercise:** The goal of this assignment is to allow you to explore and use as many of JavaScript's objects, methods, and properties as possible in a small assignment. Some functions must be written from scratch. Other functions, appropriately attributed, may be downloaded from the web and used as a part of the system or as the basis for your own functions. **PHP Exercise:** Build a set of PHP scripts that perform some dynamic server side functionality. **Understanding plug-ins:** Develop a Firefox extension.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Describe fundamental knowledge of web technologies
CLO 2	Design web applications maintaining necessary criteria
CLO 3	Implement their knowledge of server-side technologies
CLO 4	Implement their knowledge of front end technologies
CLO 5	Design web applications maintaining MVC framework
CLO 6	Develop plug-ins, APIs, extensions

CLO 7	Use Javascript, PHP and other contemporary technologies to develop web applications

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1		X	X					X	
CLO2		X	X					X	
CLO3		X	X					X	
CLO4		X	X					X	
CLO5		X	X					X	
CLO6		X	X					X	
CLO7		X	X					X	

Course Title: Communication Engineering	**Credits:** 2.0
Course No.: CSE 365	**Contact hours:** 2 hours/week

Rationale:
The goal of the course is to teach the fundamental concepts of Networking and Communication Engineering. For those interested in specializing in Communication Engineering, it provides the essentials on which later courses build.

Objectives:
- To acquaint students with different components and their respective roles in a data communication system
- To provide knowledge on layered architecture of communication protocols
- To make them understand the TCP/IP protocol suite and the OSI model
- To help them learn different encoding techniques, multiplexing techniques and switching
- To facilitate necessary knowledge on the concepts and techniques in error detection and correction

Course Contents:
Introduction: Data communications, Networks, Internet, Protocols and Standards. **Network Models:** OSI Model, TCP/IP Protocol suite, Addressing. **Data and Signals:** Analog and Digital data, Analog and Digital Signals, Time and Frequency Domain, Transmission impairments, Data rate limits, Performance. **Digital Transmission:** Digital-to-Digital Conversion, Analog-to-Digital Conversion, Transmission Modes. **Analog Transmission:** Digital-to-Analog Conversion, Analog-to-Analog Conversion. **Multiplexing and Spread Spectrum:** FDM, WDM, TDM, STDM, Digital Subscriber Line, FHSS, DSSS. **Transmission Media:** Guided and Unguided Media. **Switching:** Circuit switching, Packet switching. **Data Link Layer:** Error Detection and Correction, Data Link Control, Framing, Flow and Error Control. **Multiple Access:** CSMA, CSMA/CD, CSMA/CA, FDMA, TDMA, CDMA.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Discuss fundamental concepts of data communication and its components, different types of network, protocol layering and responsibilities of different layers for different protocols
CLO 2	Explain different types of signal related to data communication, their transmission, representation, conversion, limitations and all other relevant information
CLO 3	Identify different transmission media and their use cases
CLO 4	Understand resource sharing techniques called multiplexing for transmitting multiple signals through a single channel and identify techniques of different types of multiplexing.
CLO 5	Identify different switching techniques
CLO 6	Discuss different error detection and correction techniques and multiple access methodologies

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1		X	X		X				
CLO2		X	X		X				
CLO3		X	X		X				
CLO4		X	X		X				
CLO5		X	X		X				
CLO6		X	X		X				

Textbook
1. Data Communications and Networking by Behrouz A. Forouzan
2. Data and Computer Communications by W Stallings, Macmillan

Course Title: Communication Engineering Lab	**Credits:** 1.0
Course No.: CSE 366	**Contact hours:** 2 hours/week

Rationale:
The goal of the course is to teach the fundamental concepts of Networking and Communication Engineering. For those interested in specializing in Communication Engineering, it provides the essentials on which later courses build.

Objectives:
- To make them understand different line coding schemes
- To provide knowledge on analog transmission techniques
- To facilitate with knowledge on error detection and correction
- To provide knowledge on performance evaluation of a protocol

Course Contents:
Lab experiments and tasks:
1. Implement different Line Coding Schemes using a programming language or MATLAB.
2. Analyze different Analog Transmission Techniques using ANACOM/MODICOM.
3. Implement various Error Detection and Correction techniques using a programming language.
4. Evaluate the performance of the ARQ protocols using a Data Link Layer Protocol Simulator

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand how LAN works in a real world
CLO 2	Understand how one PC can be connected to another PC using a LAN
CLO 3	Explain how Peer-to-Peer Network works in practice
CLO 4	Discuss about different types of transmission mediums such as twisted-pair cable, coaxial cable, and fiber optic cable that are used in communication engineering
CLO 5	Understand how digital data can be transmitted to digital signals using different line coding schemes
CLO 6	Understand how Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), Amplitude Modulation (AM), Frequency Modulation (FM) and other techniques work using ANACOM/MODICOM
CLO 7	Implement different error detection and correction techniques, such as Hamming code, CRC, checksum etc.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1		X	X		X	X			
CLO2		X	X		X	X			
CLO3		X	X		X	X			
CLO4		X	X		X	X			
CLO5		X	X		X	X			
CLO6		X	X		X	X			
CLO7		X	X		X	X			

Textbook
1. Data Communications and Networking by Behrouz A. Forouzan
2. Data and Computer Communications by W Stallings, Macmillan

Course Title: Microprocessors and Interfacing	**Credits:** 3.0
Course No.: CSE 367	**Contact hours:** 3 hours/week

Rationale:
The purpose of this course is to teach students the fundamentals of microprocessor and microcontroller systems. The student will be able to incorporate these concepts into their electronic designs for other courses where control can be achieved via a microprocessor/controller implementation.

Objectives:
- To make them understand the main components and working principles of the Intel 80x86 microprocessor
- To help them develop skills on program and debug in assembly language
- To provide knowledge memory organization and memory interfacing
- To provide knowledge on hardware and software interrupts and their applications
- To provide basic ideas on designing and coding software for programmable peripheraldevices

Course Contents:

Microprocessors: Concept of microprocessor; Evolution of microprocessors; Internal architecture of Intel 8085, 8086/8088 microprocessors; Instruction set and format, Programming in machine and assembly languages, Interrupt structure, DMA, I/O operation, Microprocessor interface ICs, Peripheral interfacing, Microprocessor based system design, Coprocessor, Multiprocessor system; Intel 80286, 80386 microprocessors: memory management scheme, Protection mechanism, 80386 modes; Advanced microprocessors. Bus System: ISA, EISA, PCI AGP, Memory Bus. Centronics, SCSI, USB and GPIB standards. Interfacing with analog world: A/D conversion, digital ramp ADC, successive approximation ADC, flush ADC, tristate ADC, D/A converter, DAC specifications, DAC applications, Data acquisition, sample-and-hold circuits, Stepper Motor, Transducers, printers, motors and peripherals.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand the internal workings of an Intel 8086 microprocessor
CLO 2	Explain how computer systems are designed
CLO 3	Apply assembly language to write different programs
CLO 4	Interpret how memory is organized in systems and synchronized with system
CLO 5	Understand how a computer system is interfaced with peripheral input/output devices
CLO 6	Explain how serial and parallel communication work
CLO 7	Understand microcontroller working principles

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1				X	X				
CLO2				X	X				
CLO3				X	X				
CLO4				X	X				
CLO5				X	X				
CLO6				X	X				
CLO7				X	X				

Textbook

1. Microprocessors and Interfacing by Douglas V Hall.
2. Microprocessors and Interfacing by D. A. Godse, A.P. Godse

Course Title: Microprocessors and Interfacing Lab	**Credits:** 1.5
Course No.: CSE 368	**Contact hours:** 3 hours/week

Rationale:

The purpose of this course is to teach students the fundamentals of microprocessor and microcontroller systems. The student will be able to incorporate these concepts into their electronic designs for other courses where control can be achieved via a microprocessor/controller implementation. The theoretical knowledge is incomplete without hands-on experiments using the 8086 module and microprocessor/ microcontroller based project work.

Objectives:

- To facilitate knowledge about program and debug in assembly language
- To make them understand the memory organization and memory interfacing
- To demonstrate debugging and make understand how every instruction in 8086 works
- To interface 8086 kits with PC and program using more advanced assembler

Course Contents:

1. Registers, JMP, LOOP, CMP instruction, Conditional Jump instruction
2. Implementation of different types of instruction(rotating, shifting)
3. Instructions (MUL, IMUL, DIV, IDIV, CBW, CWD, Arrays, XLAT)
4. String instructions, macro handling
5. Bios Interrupt, Dos Interrupt
6. The IN, OUT, INS, and OUTS instruction
7. Processor signal from photodiode
8. Control of stepper motor using parallel port
9. Location detection using GPS through USB port

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Use different assembly language instructions
CLO 2	Understand 8086 microprocessor's working principle.

CLO 3	Interpret different registers and pointer values
CLO 4	Interpret the instructions executed and their effect on Flag register
CLO 5	Program different output devices
CLO 6	Design small microprocessor systems

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1				X	X	X			
CLO2				X	X	X			
CLO3				X	X	X			
CLO4				X	X	X			
CLO5				X	X	X			
CLO6				X	X	X			

Textbook

1. Microprocessors and Interfacing by Douglas V Hall.
2. Microprocessors and Interfacing by D. A. Godse, A.P. Godse

Course Title: Digital Signal Processing	**Credits:** 3.0
Course No.: CSE 325	**Contact hours:** 3 hours/week

Rationale:

Students studying CSE need to focus on fundamental concepts and techniques used in both analogue and digital signal processing (ASP and DSP) which are areas of immense interest in recent time, involving communication and biomedical signals.

Objectives:

- To provide students with fundamental knowledge and techniques in signal processing
- To help them develop skill to apply these techniques to the analysis of real world data in various fields of science
- To facilitate knowledge of how to apply the described methods in MATLAB and how to develop the algorithms based on mathematical modeling.
- To help them develop problem-solving skills associated with the application of these methods in practical, real data processing, and learn how to extract verifiable information from such applications.

Course Contents:

SIGNALS AND SYSTEMS: Basic elements of DSP – concepts of frequency in Analog and Digital Signals – sampling theorem –Discrete – time signals, systems – Analysis of discrete time LTI systems – Z transform – Convolution (linear and circular) – Correlation. **FREQUENCY TRANSFORMATIONS:** Introduction to DFT – Properties of DFT – Filtering methods based on DFT – FFT Algorithms Decimation – in – time Algorithms, Decimation – in – frequency Algorithms – Use of FFT in Linear Filtering – DCT. **IIR FILTER DESIGN:** Structures of IIR – Analog filter design – Discrete time IIR filter from analog filter – IIR filter design by Impulse invariance, Bilinear transformation, Approximation of derivatives – (HPF, BPF, BRF) filter design using frequency translation. **FIR FILTER DESIGN:** Structures of FIR – Linear phase FIR filter – Filter design using windowing techniques, Frequency sampling techniques – Finite word length effects in digital Filters. **APPLICATIONS:** Multirate signal processing – Speech compression – Adaptive filter – Musical sound processing – Image enhancement.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Explain and identify Discrete-Time Signals and Systems
CLO 2	Analyze the Fourier series and Time and Frequency domain
CLO 3	Sample signals and reconstruct signals form samples
CLO 4	Convolve and correlate signals to modify them according to need
CLO 5	Perform frequency analysis of digital signals and systems
CLO 6	Design digital filters

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X			X	X				
CLO2	X			X	X				

CLO3	X			X	X				
CLO4	X			X	X				
CLO5	X			X	X				
CLO6	X			X	X				

Textbook

1. Digital Signal Processing, Principles, Algorithms and Applications by John G. Proakis, Dimitris G. Monolakis.
2. Discrete-time Signal Processing By Alan V. Oppenheim, Ronald W. Schafer.

Course Title: Digital Signal Processing Lab	**Credits:** 1.5
Course No.: CSE 326	**Contact hours:** 3 hours/week

Rationale:

Students studying CSE need to focus on fundamental concepts and techniques used in both analogue and digital signal processing (ASP and DSP) which are areas of immense interest in recent time, involving communication and biomedical signals.

Objectives:

- To provide students with fundamental knowledge and techniques in signal processing
- To help them develop skill to apply these techniques to the analysis of real world data in various fields of science
- To facilitate knowledge of how to apply the described methods in MATLAB and how to develop the algorithms based on mathematical modeling.
- To help them develop problem-solving skills associated with the application of these methods in practical, real data processing, and learn how to extract verifiable information from such applications.

Course Contents:

SIGNALS AND SYSTEMS: Basic elements of DSP – concepts of frequency in Analog and Digital Signals – sampling theorem –Discrete – time signals, systems – Analysis of discrete time LTI systems – Z transform – Convolution (linear and circular) – Correlation. **FREQUENCY TRANSFORMATIONS:** Introduction to DFT – Properties of DFT – Filtering methods based on DFT – FFT Algorithms Decimation – in – time Algorithms, Decimation – in – frequency Algorithms – Use of FFT in Linear Filtering – DCT. **IIR FILTER DESIGN:** Structures of IIR – Analog filter design – Discrete time IIR filter from analog filter – IIR filter design by Impulse invariance, Bilinear transformation, Approximation of derivatives – (HPF, BPF, BRF) filter design using frequency translation. **FIR FILTER DESIGN:** Structures of FIR – Linear phase FIR filter – Filter design using windowing techniques, Frequency sampling techniques – Finite word length effects in digital Filters. **APPLICATIONS:** Multirate signal processing – Speech compression – Adaptive filter – Musical sound processing – Image enhancement.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Convert analog signals to digital signals
CLO 2	Sample signals and reconstruct signals from samples
CLO 3	Perform convolution and correlation on digital signals
CLO 4	Transform signals from time domain to frequency domain using Fourier transform
CLO 5	Design and implement digital filters

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X			X	X	X			
CLO2	X			X	X	X			
CLO3	X			X	X	X			
CLO4	X			X	X	X			
CLO5	X			X	X	X			

Textbook

1. Digital Signal Processing, Principles, Algorithms and Applications by John G. Proakis, Dimitris G. Monolakis.
2. Discrete-time Signal Processing by Alan V. Oppenheim, Ronald W. Schafer.

Course Title: Computer Architecture	**Credits:** 3.0
Course No.: CSE 329	**Contact hours:** 3 hours/week

Rationale:

Students wishing to build up their career in CSE need to know architecture about computers and this course will help them learn the basics of computer systems and latest hardware architectures.

Objectives:

- To make the students understand the fundamental technologies and performance evaluation of different computer systems;
- To help them know what is the instruction set architecture of a system and variations of ISA in different systems;
- To describe how computer performs arithmetic operations;
- To facilitate necessary knowledge about internal architecture of a processor;
- To provide knowledge on different levels of memory hierarchy and their management in a system.
- To accumulate basic ideas about fundamental technologies on multicore and multiprocessing system and their application

Course Contents:

Introduction to Computer Architecture: Overview and history; Cost factor; Performance metrics and evaluating computer designs. **Instruction set design:** Von Neumann machine cycle, Memory addressing, Classifying instruction set architectures, RISC versus CISC, Microprogrammed vs. hardwired control unit. **Memory System Design:** Cache memory; Basic cache structure and design; Fully associative, direct, and set associative mapping; Analyzing cache effectiveness; Replacement policies; Writing to a cache; Multiple caches; Upgrading a cache; Main Memory; Virtual memory structure, and design; Paging; Replacement strategies. **Pipelining:** General considerations; Comparison of pipelined and non pipelined computers; Instruction and arithmetic pipelines, Structural, Data and Branch hazards. **Multiprocessors and Multi-core Computers:** SISD, SIMD, and MIMD architectures; Centralized and distributed shared memory- architectures; Multi-core Processor architecture. **Input/output Devices:** Performance measure, Types of I/O device, Buses and interface to CPU, RAID. **Pipelining:** Basic pipelining, Pipeline Hazards. Parallel Processing.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Describe the fundamental technologies related to SISD, SIMD, and MIMD architectures
CLO 2	Understand and develop logic for instruction set architecture
CLO 3	Discuss organization, hierarchy and management of different levels of memory and I/O systems
CLO 4	Understand design and architecture of processor, multiprocessor, multi-core processor, distributed systems
CLO5	Differentiate between Basic Pipelining, Pipelining Hazards and Parallel Processing
CLO6	Evaluate performance of cache memory management, replacement policies and cache writing
CLO7	Analyze cache effectiveness of memory structure.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X		X						
CLO2	X		X						
CLO3	X		X						
CLO4	X		X						
CLO5	X		X						
CLO6	X		X						
CLO7	X		X						

Textbook

1. Computer Architecture and Organization by John P.Hayes.
2. Computer Organization and Design: The hardware / software interface by David A.Patterson and John L.Hennessy.

Course Title: Artificial Intelligence	**Credits:** 3.0
Course No.: CSE 337	**Contact hours:** 3 hours/week

Rationale:

Web search, speech recognition, face recognition, machine translation, autonomous driving, and automatic scheduling; these are all complex real-world problems, and the goal of artificial intelligence (AI) is to tackle these with rigorous tools. This course will help students to learn the foundational principles that drive these applications and practice implementing these systems. The main goal of the course is to equip students with the tools to tackle new AI problems they might encounter in life. This course will make students able to build applied systems and to account for intelligence from a computational point of view by introducing representations, techniques, and architectures used.

Objectives:

- To provide the most fundamental knowledge to the students so that they can understand what the AI is
- To facilitate with knowledge of theoretic proofs and formal notations of AI
- To introduces students to the basic knowledge representation, problem solving, and learning methods of artificial intelligence
- To help them develop intelligent systems by assembling solutions to concrete computational problems
- To make them understand the role of knowledge representation, problem solving, and learning in intelligent-system engineering, and appreciate the role of problem solving, vision, and language in understanding human intelligence from a computational perspective
- To help them explore applications of rule chaining, heuristic search, logic, constraint propagation, constrained search, and other problem-solving paradigms

Course Contents:

What is Artificial Intelligence: The AI problems, The underlying assumption, What is an AI technique. **Problems, Problem spaces and Search:** Defining the problem as a state space search, Production system, Problem characteristics. **Heuristics Search Techniques:** Generate and Test, Hill climbing, Best First Search, Problem Reduction, Constraint Satisfaction, Means-Ends Analysis. **Knowledge Representation Issues:** Representation and Mappings, Approaches to knowledge Representation, Issues in Knowledge representation. **Using Predicate logic:** Representing simple facts in logic, Representing Instance and Isa relationships, Computable functions and Predicates, Resolution. **Representing Knowledge using Rules:** Procedural versus Declarative Knowledge, Logic Programming, Forward versus Backward Reasoning, Matching. **Game playing:** Overview, The Minimax Search Procedure, Adding Alpha-Beta cutoffs, Additional refinements, iterative Deepening. **Planning:** Overview, An example Domain: The Blocks World, Components of a planning system, Goal stack planning. **Understanding:** What is Understanding, What makes Understanding hard, Understanding as constraint satisfaction. N**atural Language Processing:** Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing. **Expert systems:** representing and using domain knowledge, Expert system shells explanation, Knowledge Acquisition.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Compare AI with human intelligence and traditional information processing and discuss its strengths and limitations as well as its application to complex and human-centred problems.
CLO 2	Identify, analyze problems precisely and choose solution techniques using several general-purpose search techniques and heuristic methods.
CLO 3	Examine different methods for knowledge, facts and logic representation, manipulation and transformations.
CLO 4	Examine AI knowledge in a specific structured domain of game playing.
CLO 5	Construct plans to solve nontrivial problems by combining different search and decomposition techniques.
CLO 6	Understand generation and translation of knowledge in the domain of NLP
CLO 7	Design an agent to solve domain specific precise problems which usually does a human expert.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X	X						X
CLO2	X	X	X						X
CLO3	X	X	X						X
CLO4	X	X	X						X
CLO5	X	X	X						X
CLO6	X	X	X						X
CLO7	X	X	X						X

Textbook

1. Artificial Intelligence: A Modern Approach by Stuart Russel.
2. The Cambridge Handbook of Artificial Intelligence by Keith Frankish, William M. Ramsey.

Course Title: Artificial Intelligence Lab	**Credits:** 1.5
Course No.: CSE 338	**Contact hours:** 3 hours/week

Rationale:

This course is offered to help students to learn the foundational principles that drive these applications and practice implementing these systems. Course is designed with the intention to equip students with the tools to tackle new AI problems they might encounter in life. Students completing this course are supposed to build intelligent agent systems from a computational point of view.

Objectives:

- To make them understand functionality of intelligent agents
- To provide basic idea about how intelligent agents solve different real-world problems
- To make them solve different AI problem

Course Contents:
Students will have to understand the functionalities of intelligent agents and how the agents will solve general problems. Students have to use a high-level language (Python, Prolog, LISP) to solve the following problems: **Backtracking:** State space, Constraint satisfaction, Branch and bound. Example: 8-queen, 8- puzzle, Crypt-arithmetic. **BFS and production:** Water jugs problem, The missionaries and cannibal problem. **Heuristic and recursion:** Tic-tac-toe,Simple bock world, Goal stack planning, The tower of Hanoi. **Question answering:** The monkey and bananas problem.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Develop artificially intelligent systems that can deal with knowledge and logic presentation and manipulation
CLO 2	Develop artificially intelligent programs to solve brain storming puzzle games
CLO 3	Develop simple games that are artificially intelligent
CLO 4	Design, develop, examine and evaluate an intelligent agent which solves a domain specific precise problem which usually does a human expert

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1		X	X		X	X			X
CLO2		X	X		X	X			X
CLO3		X	X		X	X			X
CLO4		X	X		X	X			X

Text Books:

1. Artificial Intelligence; Elaine Rich and Kevin Knight
2. Artificial Intelligence; Winston, Patrick Henry.

Course Title: Project Work III	**Credits:** 2.0
Course No.: CSE 350	**Contact hours:** 4 hours/week

Rationale:
This is a very important course that intends to make students prepared to face challenges in relevant industry.

Objectives:

1. To facilitate necessary knowledge about latest technology
2. To help them develop skills on software development
3. To help them develop skills on teamwork and presentation

Course Contents:
Projects must possess innovative ideas which reflect contemporary IT trends. Supervisor has to ensure that every accepted project contains a basic level of research work. Projects that meet the software/hardware requirements of SUST or any other IT organization are highly preferable. Students have to give a presentation on their project works. Departments should take appropriate steps to archive all the projects and keep tracks to maintain the genuineness of the projects.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Apply latest state of the art technologies
CLO 2	Design and implement ideas for complete software
CLO 3	Evaluate existing computer and mobile applications
CLO 4	Explain ideas to groups and present their noble findings

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9

CLO1		X	X	X		X			
CLO2		X	X	X		X			
CLO3		X	X	X		X			
CLO4		X	X	X		X			

Course Title: Computer Networking	**Credits:** 3.0
Course No.: CSE 361	**Contact hours:** 3 hours/week

Rationale:
The aim of this course is to introduce key concepts and principles of computer networks to provide a solid understanding of the technologies that support modern networked computer systems. The course will use a top-down approach to study the Internet and its protocol stack. Instances of architecture, protocol, application - examples will include email, web and media-streaming. It will cover communications services (e.g., TCP/IP) required to support such network applications. The implementation and deployment of communications services in wired and wireless LAN environments will be followed by a discussion of issues of network-security and network-management. Throughout the course, the Internet's architecture and protocols will be used as the primary examples to illustrate the fundamental principles of computer networking.

Objectives:
- To provide basic knowledge about various network technologies and techniques
- To facilitate idea about the importance of layering, and the OSI reference model
- To provide knowledge of understanding of the design and operation of an IP network, such as the Internet, and explain the purpose and function of its various components
- To make them understand the general principles behind addressing, routing, reliable transmission and other stateful protocols as well as specific examples of each
- To make them be able to describe the issues in connecting heterogeneous networks

Course Contents:
Introduction: Introduction to Computer Networks, Network Goals, Applications of Networks, Network Structure, Network Architectures, The OSI Reference Model, Data Transmission in the OSI Model, OSI Terminology, The ARPANET. **Local Area Network:** LAN Technology - Architecture, Topology. **Wired LANs:** Ethernet and Fast-Ethernet, Token Ring, FDDI. **Wireless LANs:** IEEE 802.11, Bluetooth. Backbone Networks, Virtual LANs. **Wide Area Network:** SONET, Virtual Circuit Networks - Frame Relay, ATM and ATM LANs. **Network Layer:** Logical Addressing **Internet Protocol:** Internetworking, Routing Protocol, IPv4 and IPv6. **Address Mapping, Error Reporting and Multicasting:** ICMP, IGMP, ICMv6. Delivery, Forwarding and Routing. **Transport Layer:** Process-to-Process delivery, Transport Services, Protocol mechanisms, TCP, UDP, SCTP, Congestion and QoS. **Application Layer:** Domain Name System, Abstract Syntax Notation One (ASN.1), Network Management - SNMPv2, Electronic mail - SMTP and MIME, Uniform Resource Locator (URL) and Universal Resource Identifier (URI), Hypertext Transfer Protocol (HTTP). **Wireless and Mobile Networking**: Wireless Networking: Issues and Trends, Wireless Physical Layer Concepts , Wireless Cellular Networks, Mobile IP - IPv4, IPv6, TCP over Wireless, Ad Hoc Networks: Issues and Routing, Wireless Sensor Networks, Wireless Mesh and Multi-Hop Relay Networks, Wireless Network Security, Energy Management in Ad Hoc Wireless Networks..

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Identify the key components of computer networks and its architecture
CLO 2	Have an informed view of both the internal workings of the Internet and of a number of common Internet applications and protocols
CLO 3	Identify different LAN technologies and their basic principles
CLO 4	Analyze IP addressing and subnetting IP networks
CLO 5	Understand different routing protocols and algorithms
CLO 6	Analyze transport layer services, multiplexing/demultiplexing and congestion control
CLO 7	Apply knowledge of network security in different applications

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X				X	X	X	
CLO2	X	X					X	X	
CLO3	X			X	X				
CLO4	X	X	X	X	X			X	

CLO5	X		X	X	X	X		X	
CLO6	X	X	X	X	X			X	
CLO7	X			X			X	X	

Recommended Books

1. Data Communications and Networking – Behrouz A. Forouzan
2. Computer networks – A. S. Tanenbaum, Addison-Wesley.
3. Computer Networking: a Top-down Approach – James F. Kurose, Keith W. Ross

Course Title: Computer Networking Lab	**Credits:** 1.5
Course No.: CSE 362	**Contact hours:** 3 hours/week

Rationale:

The aim of this lab course is to provide practical knowledge in computer networks. Students will learn subnetting and will design a network using Packet Tracer and analysis the behavior of TCP/IP layers. The students will gain practical knowledge of configuring Switch, Router, DHCP, FTP servers.

Objectives:

- To help students designing and implementing VLSM addressing schemes in a heterogeneouscomputer network using Packet Tracer
- To make them capable to configure switch, router and end devices
- To make them capable to configure DHCP, SMTP and FTP servers
- To make them capable to design and implement a system using socket programming

Course Contents:

Subnetting and designing a network using Packet Tracer. Analysis of the TCP/IP behavior. Packet analysis. Server configuration: DHCP, SMTP, FTP, Web Switch and Router Configuration. Socket Programming.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Design and implement a heterogeneous computer network
CLO 2	Analyze the behavior of different TCP/IP layers
CLO 3	Configure DHCP, SMTP and FTP Server and test connectivity
CLO 4	Analyze IP addressing and subnetting IP networks

CLO 5	Configure switch, router and end devices in a network
CLO 6	Design a system using socket programming

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X	X			X	X	X	
CLO2	X	X		X		X			
CLO3	X	X	X	X		X	X	X	
CLO4	X	X	X	X	X	X			
CLO5	X	X	X	X		X		X	
CLO6	X	X	X	X		X	X	X	

Recommended Books

1. Data Communications and Networking – Behrouz A. Forouzan
2. Computer networks – A. S. Tanenbaum, Addison-Wesley.
3. Computer Networking: a Top-down Approach – James F. Kurose, Keith W. Ross

Course Title: Technical Writing And Presentation	**Credits:** 2.0
Course No.: CSE 376	**Contact hours:** 4 hours/week

Rationale:

In this course students will be facilitated with knowledge on interpretation of their technical knowledge through writing. They will learn how to write in a specific format using the latest technologies, draw their diagrams and also present their work in front of the audience.

Objectives:

- To facilitate necessary knowledge about methods for technical writing
- To acquaint students with basic tools for writing, presentations and drawings
- To help them enhance the skills on presentation and communication

Course Contents:

Issues of technical writing and effective oral presentation in Computer Science and Engineering; Writing styles of definitions, propositions, theorems and proofs;

Preparation of reports, research papers, theses and books: abstract, preface, contents, bibliography and index; Writing of book reviews and referee reports; Writing tools: LATEX; Diagram drawing software; presentation tools.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Apply skills on technical writing for writing technical reports
CLO 2	Use latest technologies for writing and drawing
CLO 3	Apply skills on designing graphical representations
CLO 4	Develop their presentations skills

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1				X					X
CLO2				X					X
CLO3				X					X
CLO4				X					X

Fourth Year

Course Title: Software Project Management	**Credits:** 2.0
Course No.: CSE 421	**Contact hours:** 4 hours/week

Rationale:
This course attempts to enlighten students on how software projects are managed in the current software industry and what tools are used. Many of the students who graduated from CSE will join the industry. Eventually they will be promoted to become team-lead or project manager. So, it is imperative that students develop the basic skills on how to properly manage a project (planning, scheduling, resource allocation, execution, tracking and delivery of software projects).

Objectives:
- To provide knowledge how to put together the blueprint of the entire project from ideation to fruition
- To give students an insight on how software are managed in the current industry and what tools and standards are used to manage and evaluate these projects

- To help students develop managerial skill-set such as cost estimation, risk management, configuration management, proper use of available resources etc.
- To enhance leadership skills among students so that they can assemble and lead the project team.

Course Contents:
Planning and managing of software development projects. Software process models. ISO 9000, SEI's Capability Maturity Model, continuous process improvement. Planning, scheduling, tracking, cost estimation, risk management, configuration management.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand how to put together the blueprint of the entire project from ideation to fruition.
CLO 2	Know how software are managed in the current industry
CLO 3	Identify what tools and standards are used to manage and evaluate a projects
CLO 4	Develop managerial skill-set such as cost estimation, risk management, configuration management, proper use of available resources etc.
CLO 5	Enhance leadership skills among students so that they can assemble and lead the project team

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1		X	X			X		X	X
CLO2		X	X			X		X	X
CLO3		X	X			X		X	X
CLO4		X	X			X		X	X
CLO5		X	X			X		X	X

Textbook
1. Software Project Management for Dummies by Joseph Phillips and Teresa Luckey

2. Introduction to Software Project Management by Book by Adolfo Villafiorita

Course Title: Introduction to Computer Security	**Credits:** 3.0
Course No.: CSE 461	**Contact hours:** 3 hours lab per week

Rationale:
This is an introductory course on computer security. The main objective of this course is to introduce the basic concepts of cryptography and computer security covering physical security, operating system security as well as network and web security.

Objectives:
- To facilitate the basic knowledge of classic crypto systems and basic crypto primitives
- To assist students in developing introductory knowledge about block cipher and their different modes
- To help students conceptualize basic theories of different cryptographic mechanism such as symmetric and public key encryption, digital signature and hash function
- To assist students in developing basic knowledge about different security aspects covering multiple domains such as physical security, OS security, network security and web security
- To facilitate the basic knowledge of blockchain systems

Course Contents:
Basic terminology and security concepts: Fundamental concepts, Access control models, Cryptographic concepts, Security principles **Classic Crypto Systems:** Substitution cipher, Vigenère cipher, Hill Cipher, One-time pads **Symmetric Encryption:** Advanced Encryption Standard (AES) **Public Key Encryption:** RSA and ElGamal crypto systems **Other crypto mechanisms**: Hash Function, Digital Signature **Physical security**: Authentication technologies, Direct attacks, Physical Intrusion Detection **Operating Systems Security:** Process, security, Memory and file system security, Application program security **Malware and forensic analysis:** Insider & Malware attacks, Computer viruses, Privacy-invasive software, Countermeasures, Malware forensic **Network Security:** Network security concepts, Vulnerabilities in Link, Network, Transport and Application layers, Firewall, Tunnelling and Intrusion detection, Denial of Service attacks, Countermeasures **Web security:** Attacks on clients, Attacks on servers, Countermeasures **Blockchain and Bitcoin:** History of money, The need of decentralization, State machine replication, Concepts of transaction, block, blockchain and distributed consensus of Blockchain security, Blockchain applications
Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Explain the Basic concepts of security and its different properties
CLO 2	Understand the theoretical foundation of cryptography
CLO 3	Explain the practical application of cryptography in different application domains
CLO 4	Analyze and identify security vulnerabilities in practical systems
CLO 5	Demonstrate knowledge about different tools that are used to secure different systems
CLO 6	Design different security solutions in practical systems
CLO 7	Understand the importance of human resources within an organization to maintain its security practices
CLO 8	Understand the role of users to maintain personal security

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X				X				
CLO2	X				X				
CLO3	X								
CLO4	X	X	X	X					
CLO5	X			X					
CLO6	X		X	X					
CLO7	X	X				X		X	
CLO8	X	X					X	X	

Textbooks
1. Introduction to Computer Security by Michael T. Goodrich and Roberto Tamassia
2. Introduction to Computer Security by Matt Bishop

Course Title: Introduction To Computer Security Lab	**Credits:** 1.5
Course No.: CSE 462	**Contact hours:** 3 hours/week

Rationale

In this course, these students will carry out a number of hands-on lab works based on concepts gained in its counterpart theory course, CSE 461. The main motivation of this course is to provide hands on experiences of working with different encryption algorithms, attacking systems exploiting different vulnerabilities and adopting security measures to counteract these vulnerabilities.

Course Objectives:

- To assist students in developing practical knowledge about different cryptographic algorithms
- To help students to identify different vulnerabilities within a system and to assess its security.
- To assist students in developing secure systems using different cryptographic libraries.
- To facilitate the knowledge of exploiting network vulnerabilities, attacking as well as defending a web application.

Course Contents:

Attacking classic cipher systems, Programming different cryptographic algorithms, Developing secure systems utilizing different cryptographic libraries, Exploiting network vulnerabilities, attacking and defending web applications and Malware analysis.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Explain the Basic concepts of security and its different properties
CLO 2	Understand the theoretical foundation of cryptography
CLO 3	Demonstrate practical knowledge of different types of attacks in practical systems
CLO 4	Analyze and identify security vulnerabilities in practical systems
CLO 5	Demonstrate knowledge about different tools that are used to secure different systems
CLO 6	Design and implement different security solutions in practical systems

CLO 7	Demonstrate their ability to work on teams to address a particular security problem and to design a secure system.
CLO 8	Understand different measures for the protection of personal security

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO9
CLO 1	X				X				
CLO 2	X				X				
CLO 3	X								
CLO 4	X	X	X	X					
CLO 5	X			X					
CLO 6	X		X	X					
CLO 7	X	X	X			X		X	
CLO 8	X	X					X	X	

Recommended Books

1. Introduction to Computer Security by Michae lT. Goodrich and Roberto Tamassia
2. Computer Security: Principles and Practice by William Stallings Lawrie Brown, 4th Edition
3. Introduction to Computer Security by Matt Bishop

Course Title: Machine Learning	**Credits:** 3.0
Course No.: CSE 475	**Contact hours:** 3 hours/week

Rationale:

Current digital world is totally driven by data and information. Prediction, recommendation, identification and many other activities are dependent on data mining, summarization and big data analysis. And for completing these types of works the importance of Machine Learning is increasing day by day. This course is intended to introduce students to the world where machines can learn and act based on learning like how human beings work.

Objectives:

- To familiarize with a set of well-known supervised, unsupervised and semi-supervised learning algorithms
- To help them Understand how machine learning algorithms are evaluated
- To make them able to formulate machine learning problems corresponding to different applications
- To introduce with the basic theory underlying machine learning
- To help them to apply machine learning algorithms to solve problems of moderate complexity.

Course Contents:
Introduction to Machine Learning Concepts: Concepts of ML. Types of Machine Learning, Some ML applications and examples. The main components of a ML system. Requirements to design a ML system. Testing ML algorithms, Linear Regression, Logistic Regression, Regularization, Decision Tree, Learning a concept and hypothesis, Naïve Bayes Classifier, Artificial Neural Network, Linear Discriminants, Perceptron Learning, Delta Rule, Multi-layer Neural Network, Back-propagation Algorithm, Unsupervised Learning, Clustering Technique, K-means Clustering, Clique Graph, Hierarchical Clustering, Anomaly Detection, Dimensionality Reduction, N-gram Model, Hidden Markov Model, Support Vector Machine, Genetic Algorithm, Reinforcement Learning, Information Retrieval, Natural Language Processing: Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing. Recommender System. Deep Learning.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Discuss about fundamental topics of ML, its classification, presentation, components and its impact on the current and future world.
CLO 2	Examine different types of machine learning approaches for solving different types of real life problems.
CLO 3	Understand evolution of several machine learning approaches based on their complexities and use cases.
CLO 4	Understand methods like principle component analysis, dimensionality reduction, feature extraction to prepare and manipulate data to make them suitable for ML approaches.
CLO 5	Identify a real life problem and design and develop computer based systems implementing machine learning models based on real life data from the identified domain.
CLO 6	Compare models based on different performance metrics
CLO 7	Graphically represent hidden pattern in a dataset to find out the hidden information

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X	X					X	
CLO2	X	X	X					X	
CLO3	X	X	X					X	
CLO4	X	X	X					X	
CLO5	X	X	X					X	
CLO6	X	X	X					X	
CLO7	X	X	X					X	

Textbook

1. Machine Learning, An Algorithmic Perspective(2nd Edition), Stephen Marsland
2. Introduction to machine learning (2nd edition), Alpaydin, Ethem.
3. The Art and Science of Algorithms that Make Sense of Data Machine Learning, Peter Flach.
4. Machine Learning, Tom Mitchell, McGraw Hill.

Course Title: Machine Learning Lab	**Credits:** 1.5
Course No.: CSE 476	**Contact hours:** 3 hours/week

Rationale:
Current digital world is totally driven by data and information. Prediction, recommendation, identification and many other activities are dependent on data mining, summarization and big data analysis. And for completing these types of works the importance of Machine Learning is increasing day by day. This course is intended to introduce students to the world where machines can learn and act based on learning like how human beings work.

Objectives:
- To familiarize with a set of well-known supervised, unsupervised and semi-supervised learning algorithms

- To help them Understand how machine learning algorithms are evaluated
- To make them able to formulate machine learning problems corresponding to different applications
- To help them to apply machine learning algorithms to solve problems of moderate complexity

Course Contents:
Laboratory works based on CSE475. ID3 Algorithm for Decision Tree, Regression using LSE and estimating MSE, kNN Algorithm as Nearest Neighbor Classifier, Apply NB Classifier for a Classification Task. Application of the MLP-BP ANN algorithm, Application of GA for solving a problem, Application of SVM, Application of HMM, Exclusive clustering: K- means algorithm, Agglomerative clustering: Hierarchical algorithm.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Use current tools and technologies for solving machine learning problems.
CLO 2	Examine different types of machine learning approaches for solving different types of real life problems.
CLO 3	Implement methods like principle component analysis, dimensionality reduction, feature extraction to manage data to make them suitable for ML approaches.
CLO 4	Develop computer based systems implementing machine learning models based on real life data to solve real life problems.
CLO 5	Identify a real life problem and design and develop computer based systems implementing machine learning models based on real life data from the identified domain.
CLO 6	Compare models based on different performance metrics
CLO 7	Graphically represent hidden pattern in a dataset to find out the hidden information

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X	X	X				X	
CLO2	X	X	X	X				X	
CLO3	X	X	X	X				X	
CLO4	X	X	X	X				X	
CLO5	X	X	X	X				X	
CLO6	X	X	X	X				X	
CLO7	X	X	X	X				X	

Textbook
1. Machine Learning, An Algorithmic Perspective(2nd Edition), Stephen Marsland
2. Introduction to machine learning (2nd edition), Alpaydin, Ethem.
3. The Art and Science of Algorithms that Make Sense of Data Machine Learning, Peter Flach.
4. Machine Learning, Tom Mitchell, McGraw Hill.

Course Title: Thesis I	**Credits:** 2.0
Course No.: CSE 450	**Contact hours:** 4 hours/week

Rationale:
This course is intended to introduce students with research work. Some criteria will be set by the department which will be used to check whether a student is eligible or not for this course. Those criteria may change according to the decision of the responsible committee. In this course students will do some research work to find out contemporary important topics related to their field of interest and submit proposals to continue further research work to find out something noble.

Objectives:
- To help them learn how to read research articles
- To facilitate basic idea about research methodologies
- To provide knowledge about research proposal
- To help to develop skill of presentation

Course Contents:
Project work based on all major courses.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Find out necessary articles from the vast source of research world
CLO 2	Understand how to read an article efficiently

CLO 3	Understand how a research work should be done
CLO 4	Explore the research world to find out to work on something noble and important
CLO 5	Create and submit their research proposal
CLO 6	Present and defend their idea in front of audience

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X				X		X	
CLO2	X	X				X		X	
CLO3	X	X				X		X	
CLO4	X	X				X		X	
CLO5	X	X				X		X	
CLO6	X	X				X		X	

Course Title: Project I	**Credits:** 2.0
Course No.: CSE 480	**Contact hours:** 4 hours/week

Rationale:
This course is offered as the alternative of Thesis I. Students who will be interested to implement their knowledge on development may select a project and will develop full functioning software projects.

Objectives:
- To facilitate necessary knowledge about latest technology
- To help them develop skills on software development
- To help them develop skills on teamwork and presentation

Course Contents:
Project work based on all major courses.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Apply latest state of the art technologies
CLO 2	Design and implement ideas for complete software
CLO 3	Evaluate existing computer and mobile applications
CLO 4	Explain ideas to groups and present their noble findings

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1		X	X	X		X		X	
CLO2		X	X	X		X		X	
CLO3		X	X	X		X		X	
CLO4		X	X	X		X		X	

Course Title: Computer Graphics	**Credits:** 3.0
Course No.: CSE 493	**Contact hours:** 3 hours/week

Rationale:
In many engineering applications (e.g. automotive, aerospace, medical), the ability to quickly visualize newly designed shapes is indispensable. Using computer graphics, designers can interactively view and modify models of their shapes using a computer. Therefore, a student who is willing to build his/her career in modeling and visualizing the data from imaging this course will help them to learn the fundamentals and tools used to create and manipulate digital graphics.

Objectives:
- To provide knowledge on the basic elements and skills involved in the creation of computer graphics
- To help them to learn how to apply computer graphics skills and capacities to enhance published content
- To facilitate knowledge about how to model and visualize different products, buildings and cars etc. and visualize data from medical imaging such as CT scans
- To help them learn about the connection between computer graphics capacities and skills and workplace career and professional opportunities

Course Contents:

Computer Graphics Programming: OpenGL. **Camera Analogy:** Viewing, Windowing, Clipping. **Projective Transformation (Ray-tracing):** Orthogonal Projection, Perspective Projection. **Vector:** Normal Vector, View Vector. **Matrix:** 2D and 3D Rotation and Translation Matrix. **Raster Graphics:** Line Drawing, Anti-aliasing, Polygon Filling Algorithms. **Hidden Surface Removal:** z-buffering. **Lighting and Surface Property:** Diffused Light, Ambient Light, Specular Light, Lighting Models for reflection. **Shading:** Flat Shading, Lambert Shading, Phong Shading. **Texture Mapping:** Texture Fundamentals. **Animation:** Real time animation.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Explain basic concepts of computer graphics, viewing, windowing, clipping.
CLO 2	Explain and apply the algorithms : line drawing, polygon filling, clipping and transformation and orthogonal and perspective projections
CLO 3	Perform 3D transformations, viewing and clipping
CLO 4	Apply algorithms related to hidden surface removal that includes but not limited to Z-buffer algorithm, the Painter's algorithm
CLO 5	Create and adjust smooth motions of objects in a scene.
CLO 6	Predict collisions between simple moving objects (points, lines, balls...) and how to simulate elastic shocks between them.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X		X						
CLO2	X		X						
CLO3	X		X						
CLO4	X		X						
CLO5	X		X						
CLO6	X		X						

Textbook

1. Computer Graphics: Principles and Practice, Folley, Van Damm, Feiner, Hughes,

2. Computer Graphics: A Programming Approach: Steven and Harrington.

Course Title: Computer Graphics Lab	**Credits:** 1.5
Course No.: CSE 494	**Contact hours:** 3 hours/week

Rationale:

In many engineering applications (e.g. automotive, aerospace, medical), the ability to quickly visualize newly designed shapes is indispensable. Using computer graphics, designers can interactively view and modify models of their shapes using a computer. Therefore, a student who is willing to build his/her career in modeling and visualizing the data from imaging this course will help them to learn the fundamentals and tools used to create and manipulate digital graphics.

Objectives:

- To acquaint the students with the implementation of fundamental algorithms in Computer Graphics.

Course Contents:

Tool to use for lab: OpenGL

1. Line Drawing: Bresenhams
2. Region Filling: Scan Line Algorithm
3. Transformation: 2D and 3D translation, Rotation, Scaling
4. Clipping: Line and Polygon
5. Projection: Perspective and Parallel
6. Animation: Morphing

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Use OpenGL
CLO 2	Create interactive graphics applications in C++ using one or more graphics application programming interfaces.
CLO 3	Draw basic geometric shapes (Points, Lines, Circles) using drawing algorithms.
CLO 4	Apply geometrical transformations on graphical problem solving.
CLO 5	Develop skill to generate computer graphics animation software.
CLO 6	Demonstrate 2D and 3D graphics processing techniques. (transformation, viewing, clipping)

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X		X	X					
CLO2	X		X	X					
CLO3	X		X	X					
CLO4	X		X	X					
CLO5	X		X	X					
CLO6	X		X	X					

Textbook

1. Computer Graphics: Principles and Practice, Folley, Van Damn, Feiner, Hughes,
2. Computer Graphics: A Programming Approach: Steven and Harrington.

Course Title: Thesis II	**Credits:** 4.0
Course No.: CSE 452	**Contact hours:** 8 hours/week

Rationale:

Students who selected CSE 450 will continue their proposed research work in this course. In this course students are supervised to complete their research and find out something new that can create impact on the contemporary research world.

Objectives:

- To help them learn how to read research articles
- To facilitate basic idea about research methodologies
- To provide knowledge on writing thesis reports
- To help to develop skill of presentation

Course Contents:

Project work based on all major courses.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Find out necessary articles from the vast source of research world
CLO 2	Understand how to read an article efficiently
CLO 3	Understand how a research work should be done
CLO 4	Explore the research world to find out to work on something noble and important
CLO 5	Create report based on their findings
CLO 6	Present and defend their hypothesis in front of audiences

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X				X		X	
CLO2	X	X				X		X	
CLO3	X	X				X		X	
CLO4	X	X				X		X	
CLO5	X	X				X		X	
CLO6	X	X				X		X	

Course Title: Project II	**Credits:** 4.0
Course No.: CSE 482	**Contact hours:** 8 hours/week

Rationale:

Students who selected Thesis in CSE 480 will continue their work in CSE 482 and complete their project.

Objectives:

- To facilitate necessary knowledge about latest technology
- To develop skills on software development
- To develop skills on teamwork and presentation

Course Contents:

Project work based on all major courses.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Apply latest state of the art technologies

CLO 2	Design and implement ideas for complete software
CLO 3	Evaluate existing computer and mobile applications
CLO 4	Explain ideas to groups and present their noble findings
CLO 5	Create report based on their final outcome
CLO 6	Present and defend their project in front of audience

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1		X	X	X		X		X	
CLO2		X	X	X		X		X	
CLO3		X	X	X		X		X	
CLO4		X	X	X		X		X	
CLO5		X	X	X		X		X	
CLO6		X	X	X		X		X	

Course Title: Viva Voce	**Credits:** 1.0
Course No.: CSE 484	**Contact hours:** 2 hours/week

Rationale:
This course objects to prepare students for their upcoming real life interviews.

Objectives:
- To make students get mentally prepared for real life interviews
- To make them recall all the important and fundamental knowledge they have acquired during the full undergrad session

Course Contents:
Viva based on studied major courses.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Present skills on facing verbal sessions.
CLO 2	Recall fundamental information they acquired in their undergrad life
CLO 3	Argue logically and defend their answer

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X								X
CLO2	X								X
CLO3	X								X

OPTIONAL COURSES

Course Title: Compiler Construction	**Credits:** 3.0
Course No.: CSE 439	**Contact hours:** 3 hours/week

Rationale:
Students wishing to build up their career in CSE need to develop software using programming language and this course will help them learn the internal mechanism of compiler technologies and make them better programmers and increase their ability to learn new programming languages quickly.

Objectives:
- To provide knowledge about the structure of compilers;
- To acquaint with basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, intermediate code generation, optimization, and code generation;
- To acquaint with basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines;
- To help to design and implement a simple compiler using a software engineering approach;

Course Contents:

Introduction to compilers: Introductory concepts, types of compilers, applications, phases of a compiler. Lexical analysis: Role of the lexical analyzer, input buffering, token specification, recognition of tokens, symbol tables. Parsing: Parser and its role, context free grammars, top-down parsing. Syntax-directed translation: Syntax-directed definitions, construction of syntax trees, top-down translation. Type checking: Type systems, type expressions, static and dynamic checking of types, error recovery. Run-time organization: Run-time storage organization, storage strategies. Intermediate code generation: Intermediate languages, declarations, assignment statements. Code optimization: Basic concepts of code optimization, principal sources of optimization. Code generation. Features of some common compilers: Characteristic features of C, Pascal and Fortran compilers.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand the structure of compilers.
CLO 2	Understand lexical and syntax analyzers and use them in the construction of scanners and parsers
CLO 3	Understand the basic data structures used in compiler construction such abstract syntax trees, symbol tables, three-address code and stack machines.
CLO 4	Discuss intermediate code generation and code optimization techniques.
CLO 5	Examine features of some common compilers.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1			X		X				
CLO2			X		X				
CLO3			X		X				
CLO4			X		X				
CLO5			X		X				

Textbook

1. Compilers: Principles, Techniques, and Tools – Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman.
2. Engineering a Compiler - Keith Cooper, Linda Torczon.

Course Title: Compiler Construction Lab	**Credits:** 1.5
Course No.: CSE 440	**Contact hours:** 3 hours/week

Rationale:

Students wishing to build up their career in CSE needs developing software using programming language and this course will help them learn the internal mechanism of compiler technologies and make them better programmer and increase their ability to learn new programming languages quickly.

Objectives:

- To provide knowledge about the structure of compilers;
- To acquaint with basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, intermediate code generation, optimization, and code generation;
- To acquaint with basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines;
- To help to design and implement a simple compiler using a software engineering approach;

Course Contents:

How to use scanner and parser generator tools (e.g., Flex, JFlex, CUP, Yacc, etc). For a given simple source language designing and implementing lexical analyzer, symbol tables, parser, intermediate code generator and code generator.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Use generators like Lex/Flex/JFlex and CUP/Yacc
CLO 2	Use lexical and syntax analyzers and use them in the construction of scanners and parsers
CLO 3	Implement the basic data structures used in compiler construction such abstract syntax trees, symbol tables, three-address code and stack machines.
CLO 4	Design and implement a simple compiler using a software engineering approach;
CLO 5	Examine features of some common compilers

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO 1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9

CLO1			X	X	X				
CLO2			X	X	X				
CLO3			X	X	X				
CLO4			X	X	X				
CLO5			X	X	X				

Textbook

1. Modern Compiler Implementation in Java/C, Andrew W. Appel.

Course Title: Bioinformatics	**Credits:** 3.0
Course No.: CSE 469	**Contact hours:** 3 hours/week

Rationale:

In this course, students will learn fundamental concepts and methods in bioinformatics. This course will provide a certain level of understanding of molecular biology and a working knowledge of bioinformatics applications and databases covering the topics sequence similarity and alignments, evolutionary processes, protein structure, genome characteristics and gene expression.

Objectives:

- To introduce students to the fundamentals of evolution, molecular biology and molecular evolution.
- To show students how to apply many of the basic predictive methods that are common in modern bioinformatics.
- To make the students have a practical and hands-on experience with common bioinformatics tools and databases.
- To train the students in the basic theory and application of programs used for database searching, protein and DNA sequence analysis, prediction of protein function, and building phylogenetic trees.

Course Contents:

Cell concept: Cell to Chromosome, **Cell division. Nucleic acids:** Structure and properties of different forms of DNA and RNA; DNA replication. **Proteins:** Structure and classification, Central dogma of molecular biology. **Genetic code:** A brief account. **Genetics:** Mendel's laws of inheritance, Organization of genetic material of prokaryotes and eukaryotes, repetitive DNA, chromosome organization and banding patterns, structure of gene - intron, exon and their relationships, overlapping gene, regulatory sequence, Molecular mechanism of general recombination, gene conversion, Evolution and types of mutation, molecular mechanisms of mutation. **Introduction to Bioinformatics:** Definition and History of Bioinformatics, Bioinformatics Tools and Databases, Applications of Bioinformatics. **Sequence alignment:** Dynamic programming. Global, local, semiglobal. Scoring matrices. The Blast family of programs. Significance of alignments, Aligning more than two sequences. Patterns, Profiles and Multiple Alignments, Genomes alignment. Structure-based alignment. **Hidden Markov Models in Bioinformatics:** Definition and applications in Bioinformatics. Examples of the Viterbi, the Forward and the Backward algorithms. Parameter estimation for HMMs. **Trees:** The Phylogeny problem. Distance methods, parsimony, bootstrap. Stationary Markov processes. Rate matrices. Maximum likelihood. Felsenstein's post-order traversal. **Finding regulatory elements,** Gibbs sampling. **Gene Detection and Genome Annotation, Gene Expression Analysis.**

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand basic concepts of Biology and Bioinformatics and its significance in Biological Data analysis.
CLO 2	*Describe the history, scope and importance of Bioinformatics.*
CLO 3	Explain about the methods to characterize and manage the different types of Biological data.
CLO 4	*Discuss the basics of sequence alignment and analysis.*
CLO 5	*Use tree data structure for genome sequencing.*

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X				X				
CLO2	X				X				
CLO3	X				X				
CLO4	X				X				
CLO5	X				X				

Textbook

1. An Introduction to Bioinformatics Algorithms by Neil C. Jones and Pavel A. Pevzner.
2. Understanding Bioinformatics by Market Zvelebil, Jeremy O. Baum
3. Biological Sequence Analysis
4. Bioinformatics for Biologists by Pavel Pevzner and Ron Shamir.

Course Title: Bioinformatics Lab	**Credits:** 1.5
Course No.: CSE 470	**Contact hours:** 3 hours/week

Rationale:
In this course, students will learn fundamental concepts and methods in bioinformatics. This course will provide certain level of understanding of molecular biology and a working knowledge of bioinformatics applications and databases covering the topics sequence similarity and alignments, evolutionary processes, protein structure, genome characteristics and gene expression.

Objectives:
- To introduce students to the fundamentals of evolution, molecular biology and molecular evolution.
- To show students how to apply many of the basic predictive methods that are common in modern bioinformatics.
- To make the students have a practical and hands-on experience with common bioinformatics tools and databases.
- To train the students in the basic theory and application of programs used for database searching, protein and DNA sequence analysis, prediction of protein function, and building phylogenetic trees.

Course Contents:
Cell concept: Cell to Chromosome, **Cell division. Nucleic acids:** Structure and properties of different forms of DNA and RNA; DNA replication. **Proteins:** Structure and classification, Central dogma of molecular biology. **Genetic code:** A brief account. **Genetics:** Mendel's laws of inheritance, Organization of genetic material of prokaryotes and eukaryotes, repetitive DNA, chromosome organization and banding patterns, structure of gene - intron, exon and their relationships, overlapping gene, regulatory sequence, Molecular mechanism of general recombination, gene conversion, Evolution and types of mutation, molecular mechanisms of mutation. **Introduction to Bioinformatics:** Definition and History of Bioinformatics, Bioinformatics Tools and Databases, Applications of Bioinformatics. **Sequence alignment:** Dynamic programming. Global, local, semiglobal. Scoring matrices. The Blast family of programs. Significance of alignments, Aligning more than two sequences. Patterns, Profiles and Multiple Alignments, Genomes alignment. Structure-based alignment. **Hidden Markov Models in Bioinformatics:** Definition and applications in Bioinformatics. Examples of the Viterbi, the Forward and the Backward algorithms. Parameter estimation for HMMs. **Trees:** The Phylogeny problem. Distance methods, parsimony, bootstrap. Stationary Markov processes. Rate matrices. Maximum likelihood. Felsenstein's post-order traversal. **Finding regulatory elements,** Gibbs sampling. **Gene Detection and Genome Annotation, Gene Expression Analysis.**

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand basic concepts of Biology and Bioinformatics and its significance in Biological Data analysis.
CLO 2	*Describe the history, scope and importance of Bioinformatics.*
CLO 3	Explain about the methods to characterize and manage the different types of Biological data.
CLO 4	*Discuss the basics of sequence alignment and analysis.*
CLO 5	*Use tree data structure for genome sequencing.*

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X				X	X			
CLO2	X				X	X			
CLO3	X				X	X			
CLO4	X				X	X			
CLO5	X				X	X			

Textbook
1. An Introduction to Bioinformatics Algorithms by Neil C. Jones and Pavel A. Pevzner.
2. Understanding Bioinformatics by Market Zvelebil, Jeremy O. Baum
3. Biological Sequence Analysis
4. Bioinformatics for Biologists byPavelPevzner and Ron Shami

Course Title: Advanced Database System	**Credits:** 3
Course No.: CSE 455	**Contact hours:** 3 hours/week

Rationale:
The course explores advanced database systems, their management and their corporate role. At the heart of information systems lie database management systems, transactional database systems, data warehouses and databases for storing complex data. This course looks at the technologies, data models and policies that such systems require.

Objectives:

- To facilitate in depth information about query process and optimization.
- To make students understand and apply transaction and concurrency control.
- To provide the knowledge of non-relational and spatial databases.
- To help to develop an understanding of essential data mining concepts.
- To facilitate the basic concepts and algorithms of data warehousing.

Course Contents:
Query Processing and Optimization: Query Interpretation, Equivalence of Expressions, Estimation of Query-ProcessingCost, Estimation of Costs of Access Using Indices, Join Strategies, Join Strategies for parallel Processing, Structure of the query Optimizer, Transformation of Relational Expression, Rewrite parse Tree **Transactions and Concurrency Control:** Schedules, Testing for Serializability, Lock-Based Protocols, Timestamp-BasedProtocols, Validation Techniques, Multiple Granularity, Multiversion Schemes, Insert and Delete Operations, Deadlock Handling. **NoSQL:** Scalability, CAP Theorem, BASE System, ACID vs BASE. **Spatial Database:** Object Relational Model, Spatial data, Geometry types, Data Model – (Element, Geometry, Layer), Coordinate System, Tolerance, R-Tree etc. **Data Mining:** Type of Data, Type of Interestingness, Data Mining vs Statistical Interference, Data Preprocessing, Types ofAttributes. **Data mining Concepts:** Association Rule Mining (Apriori Algorithm), Classification (Decision Tree, Support Vector Machine, Naïve Bayes Classifier), Clustering – (K-means with variations, KNN, Genetic Algorithm) etc. **Data Warehousing:** Basic concepts and algorithms.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Analyze the background processes involved in queries and transactions, and explain how these impact on database operation and design
CLO 2	Process and optimize queries.
CLO 3	Design systems that control concurrent schedules..
CLO 4	Differentiate between relational and non-relational databases and decide when to use what.
CLO 5	Use spatial databases.
CLO 6	Discuss the essential data mining concepts.
CLO 7	Explain the concepts of data warehousing.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
PLO									
CLO1	X			X					
CLO2	X	X		X	X				
CLO3	X	X	X	X	X				
CLO4	X			X					
CLO5	X			X					
CLO6	X			X					
CLO7	X			X					

Textbook
1. Introduction to Data Mining – Pang-Ning Tan, Michael Steinbach, Vipin Kumar.
2. Advanced Database Systems – Carlo Zaniolo et al (The Morgan Kaufmann Series).
3. Oracle Advanced PL/SQL Programming with CD-ROM- ScottUrman.

Course Title: Advanced Database System Lab	**Credits:** 1.5
Course No.: CSE 456	**Contact hours:** 3 hours/week

Rationale:
Advanced Database System LAB course will concentrate on the design and implementation of NoSQL and Spatial Databases and applying queries.

Objectives:
- To help students understand the different issues involved in the design and implementation of non-relational and spatial databases.
- To help them implement NoSQL and Spatial databases.
- To help them write optimized queries.
- To help them implement data mining and data warehousing algorithms.

Course Contents:
Laboratory works based on theory classes including NoSQL and Spatial Database.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Design and implement NoSQL and Spatial databases.

CLO 2	Write optimized queries.
CLO 3	Implement data mining algorithms.
CLO 4	Implement basic algorithms of data warehousing.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X		X	X					
CLO2	X	X	X	X					
CLO3	X	X		X					
CLO4	X	X		X					

Textbook

1. Oracle Spatial User's Guide and Reference, 10g Release 1 – Chuck Murray.

More Optional Courses

CSE453 (3 Credits) **CLOUD COMPUTING**

3 Hours/Week

Introduction to different types of computing: Edge computing, Grid computing, Distributed Computing, Clustercomputing, Utility computing, Cloud computing. **Cloud computing architecture:** Architectural framework; Cloud deployment models; Virtualization in cloud computing; Parallelization in cloud computing; Green cloud. Cloud Bus; **Cloudservice models**: Software as a Service (SaaS); Infrastructure as a Service (IaaS); Platform as a Service (PaaS). **Foundational elements of cloud computing**: Virtualization; Cloud computing operating System; Browser as a platform;Advanced web technologies (Web 2.0, AJAX and Mashup); Introduction to autonomic systems; Service Level Agreements(SLA); Security/Privacy; Cloud economics; Risks assessment; Current challenges facing cloud computing.**Case studies.**

Textbook

1. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things- Kai Hwang, Jack Dongarra, Geoffrey C. Fox.
2. Cloud Computing, Principles , System and Applications- Antonopoulos, Nikos, Gillam, Lee.
3. Cloud Computing: From Parallel Processing to the Internet of Things- Kai Hwang, Jack Dongarra, Geoffrey C. Fox.

4. Cloud Computing, Principles , System and Applications- Antonopoulos, Nikos, Gillam, Lee.

CSE454(1.5 Credits) CLOUD COMPUTING LAB

3 Hours/Week

Creating Windows servers on the cloud; Creating Linux servers on the cloud; Deploying applications on the cloud; Major cloud solutions and troubleshootings.

Textbook

Oracle Spatial User's Guide and Reference, 10g Release 1 – Chuck Murray.

CSE457(3 Credits) MOBILE AND WIRELESS COMMUNICATION

3 Hours/Week

Aspects of radio wave propagation for fixed and mobile communication systems, and cellular system design. Large-scale and small-scale propagation models, multipath fading, link-budget, interference and frequency reuse, multiple access schemes and system capacity. Trunking and grade of service, wireless network planning and operation. Architecture and operation of 2G cellular mobile systems, 2.5 G and 3G technologies. Special techniques/Diversity, Equalization, Interleaving, and Smart Antenna.

Textbook

1. Mobile and Wireless Communications: An Introduction- Gordon A. Gow, Richard K. Smith.
2. Communications Engineering: Essentials for Computer Scientists and Electrical Engineers- Richard Chia Tung Lee, Mao-Ching Chiu, Jung-Shan Lin

CSE458 (1.5 Credits)MOBILE AND WIRELESS COMMUNICATION LAB

3 Hours/Week

Laboratory works based on theory classes

CSE459(3 Credits) ADVANCED DATA STRUCTURE AND ALGORITHM

3 Hours/Week

Red-Black Tree, Binary Index Tree, Segment Tree, Range minimum query, lowest common ancestor, k-d Tree, Interval tree, R-tree. Advanced Application of Dynamic Programming and Backtracking.**Advanced String Structure and algorithm:** tree, suffix tree, suffix array, Aho-Corasic.**Computational Geometry:** Line Sweeping algorithms, Binary Space Partition Trees and Painter's algorithm (otheradvanced computational geometry).**Optimization of network flow:** Dinic's algorithm, Hungarian algorithm, Min cost max flow, min cut, graph coloring.Genetic algorithm and its different applications, Basic Game theory, Linear programming, Polynomials and Fast Fourier Transform, Encryption and Decryption.

CSE460 (1.5 Credits**)** ADVANCED DATA STRUCTURE AND ALGORITHM LAB

3 Hours/Week

Red-Black Tree, K-d Tree, Suffix Tree, Suffix Array, Line Sweeping algorithms, Painter's algorithm, Hungarian algorithm, Dinic's algorithm, Min cost max flow and the selected problem assign by the corresponding instructor.

CSE463 (3 Credits) SECURITY ENGINEERING
3 Hours/Week
This course provides an introduction to the new area of Security Engineering, and provides examples drawn from recent research. This is a multi-disciplinary field combining technical aspects of Applied Cryptography, Computer Engineering, and Networking as well as issues from Psychology, Policy and Economics.
Security Protocols. Identity & Authentication. Authorisation. Trust Management. Application of Economics in Security.Secure Systems Development.System Evaluation and Assurance. Secure Monitoring.Privacy Issues. Security and privacy by design paradigm.Security models and polices. Security Usability. Blockchain engineering.

Textbook
1. Security Engineering: A Guide to Building Dependable Distributed Systems – Ross Anderson
2. Cryptography Engineering: Design Principles and Practical Applications– Niels Ferguson , Bruce Schneier , Tadayoshi Kohno

CSE464 (1.5 Credits) SECURITY ENGINEERING LAB
3 Hours/Week
Laboratory works based on CSE 463.

CSE465(3 Credits) INTRODUCTION TO INTERNET OF THINGS
3 Hours/Week
Introduction to IoT. Sensing, Actuation, Networking, Communication Protocols, Sensor Networks, Machine-to-Machine Communications.
Interoperability in IoT, Introduction to Arduino Programming, Integration of Sensors and Actuators with Arduino.
Introduction to Python programming, Introduction to Raspberry Pi, Implementation of IoT with Raspberry Pi, Data Handling and Analytics, Cloud Computing, Fog Computing, Connected Vehicles, Smart Grid, Industrial IoT.

Textbook
1. The Internet of Things: Enabling Technologies, Platforms, and Use Cases by Pethuru Raj and Anupama C. Raman (CRC Press).
2. Internet of Things: A Hands-on Approach by ArshdeepBahga and Vijay Madisetti (Universities Press).

CSE466 (1.5 Credits) INTRODUCTION TO INTERNET OF THINGS LAB
3 Hours/Week
Laboratory works based on CSE 465.

CSE467(3 Credits) VLSI DESIGN

3 Hours/Week
VLSI design methodology: top-down design approach, technology trends. NMOS, CMOS inverters, pass transistor and pass gates: dc and transient characteristics. Brief overview of fabrication process: NMOS, CMOS, Bi-CMOS process. NMOS and CMOS layout, stick diagram and design rules. CMOS circuit characteristics and performance estimation: resistance and capacitance, rise and fall time, power estimation. Buffer circuit design. Introduction to Bi-CMOS circuits.Complex CMOS gates. CMOS building block: multiplexer, barrel shifter, adder, counter, multipliers. Data Path and memory structures. Design style: FPGA and PLDs.Introduction to HDL: basic digital design using VHDL.

Textbook
1. Basic VLSI Design - Douglas A. Pucknell
2. CMOS VLSI Design: A Circuits and Systems Perspective – Neil Weste, David Harris.

CSE468(1.5 Credits) VLSI DESIGN LAB
3 Hours/Week
Laboratory works based on theory classes.

CSE471(3 Credits) DIGITAL IMAGE PROCESSING
3 Hours/Week

Image Fundamentals: Introduction**,** Steps in Digital Image Processing, Components, Elements of Visual Perception, Image Sensing and Acquisition, Image Sampling and Quantization, Relationships between pixels, Color models.
Image Enhancement: Background, Enhancement by Point Processing, **Spatial Domain:** Gray level transformations, Histogram processing, Basics of Spatial Filtering, Smoothing and Sharpening Spatial Filtering, **Frequency Domain:** Introduction to Fourier Transform, Smoothing and Sharpening frequency domain filters, Ideal, Butterworth and Gaussian filters.
Image Restoration: Noise Models – Mean Filters, Order Statistics, Adaptive Filters, Band Reject Filters, Band Pass Filters, Notch Filters, Optimum Notch Filtering, Inverse Filtering, Wiener Filtering.
Image Segmentation: Detection of Discontinuities–Edge Linking and Boundary detection – Region based segmentation- Morphological processing- erosion and dilation.
Image Compression: Fundamentals, Image Compression Models, Error Free Compression, Variable Length Coding, Bit-Plane Coding, Lossless Predictive Coding, Lossy Compression, Lossy Predictive Coding, Compression Standards.

Textbook
1. Rafael C. Gonzales, Richard E. Woods, "Digital Image Processing", Third Edition, Pearson Education, 2010

References

1. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, "Digital Image Processing Using MATLAB", Third Edition Tata Mc Graw Hill Pvt. Ltd., 2011.
2. Anil Jain K. "Fundamentals of Digital Image Processing", PHI Learning Pvt. Ltd., 2011.

CSE472(1.5 Credits) DIGITAL IMAGE PROCESSING LAB
3 Hours/Week
Laboratory course is based on the theory course CSE 471. Matlab Image processing toolbox can be used to implement different image processing algorithms.

CSE473(3 Credits) NATURAL LANGUAGE PROCESSING
3 Hours/Week

Introduction. Words: Regular Expressions and Automata, Words and Transducers, N-Grams, Parts-of-Speech Tagging,Hidden Markov and Maximum Entropy Models, **Syntax:** Formal Grammars, Syntactic Parsing, Statistical Parsing, Features and Unification, Language and Complexity. **Semantics and Pragmatics:** The Representation of Meaning, Computational Semantics, Lexical Semantics, Computational Lexical Semantics, Computational Discourse. **Applications:** Information Extraction, Question Answering and Summarization, Dialogue and Conversational Agents, Machine Translation.

Textbook
1. J. H. Speech and Language Processing, Jurafsky, D. and Martin.
2. Foundations of Statistical Natural Language Processing, Manning, C. D. and H. Schütze.

CSE474 (1.5 Credits) NATURAL LANGUAGE PROCESSING LAB
3 Hours/Week
Morphological Analysis, N-grams Modeling, POS Tagging, Syntactic Parsing, Statistical Parsing, Semantic and Pragmatic Analysis.

CSE477 (3 Credits)
CONTEMPORARY COURSE ON COMPUTER SCIENCE AND ENGINEERING I (3 Hours/Week)
This course covers a contemporary title in Conputer science and Engineering decided by the department.

CSE478 (3 Credits)
CONTEMPORARY COURSE ON COMPUTER SCIENCE AND ENGINEERING I LAB (3 Hours/Week)
Laboratory works based on CSE 477.

CSE487(3 Credits)
CONTEMPORARY COURSE ON COMPUTER SCIENCE AND ENGINEERING II (3 Hours/Week)

This course covers a contemporary title in Conputer science and Engineering decided by the department.

CSE488 (1.5 Credits)
CONTEMPORARY COURSE ON COMPUTER SCIENCE AND ENGINEERING II LAB (3 Hours/Week)
Laboratory works based on CSE 479.

CSE485 (3 Credits) PARALLEL AND DISTRIBUTED COMPUTING
3 Hours/Week
Introduction: Scope, issues, applications and challenges of Parallel and Distributed Computing; Parallel and Distributed Architectures.
Parallel Computing: Parallel Performance; Shared Memory and Threads; Parallel Algorithms; OpenMP; Scalable Algorithms; Message Passing;MPI; Grid Computing.
Distributed Computing: Distributed Systems; MapReduce; Clusters; Distributed Coordination; Distributed Consensus; Distributed File Systems (DFS); Distributed Shared Memory; Distributed Transactions and Replication; Applications of Distributed Computing in Security.
Cloud Computing: Cloud Architectural Framework; Cloud Deployment Models: Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS); Cloud Economics; Service Level Agreements(SLA); Security and Privacy issues in Cloud Computing.

Textbook
1. Distributed Systems: Concepts and Designs - George Coulouris,JeanDollimore, Tim Kindberg and Gordon Blair
2. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things- Kai Hwang, Jack Dongarra, Geoffrey C. Fox.
3. Principles of Parallel Programming - Calvin Lin, Larry Snyder
4. Cloud Computing, Principles , System and Applications- Antonopoulos, Nikos, Gillam, Lee.

CSE486 (1.5Credits) PARALLEL AND DISTRIBUTED COMPUTING LAB
3 Hours/Week
Lab sessions will be based on theoretical contents.

CSE495 (3 credits) HUMAN COMPUTER INTERACTION
3 Hours/Week
Introduction: Fundamentals concepts, historical background, User Interface paradigms. Basics of human perception: Perceiving color, pattern perception, cognition. Usability Goals and Design Principles: Usability goals, user experience goals, design principles. Users: User profiles, types of users, stakeholders. Requirements: Types of requirements, requirements gathering methods, choosing a method for gathering needs. Scenarios and Tasks: Creating scenarios, tasks decompositions, hierarchical task analysis, hierarchical task graphs. Prototyping: low-fidelity prototyping, high-fidelity prototyping, prototyping dimensions, usability criteria, scenario-to-t-to-prototype. Implementations: Layout hierarchies, layout

models, visual inheritance, events and window systems, event handling, input events, paint event, image transformations, user interactions. Usability evaluation: User evaluation, evaluation approaches, ethics, data collection techniques, cognitive walkthroughs, heuristic evaluation, observing users, experiments, user testing and usability testing

Textbook:

1. Jenny Preece, Yvonne Rogers and Helen Sharp. Interaction design: Beyond Human-Computer Interaction, Wiley and sons, 2nd Edition.
2. Dix A. et al., Human-Computer Interaction. Harlow, England: Prentice Hall, 2004, ISBN-10: 0130461091.
3. Yvonne Rogers, Helen Sharp, Jenny Preece, Interaction Design: Beyond Human Computer Interaction, 3rd Edition, Wiley, 2011, ISBN-10: 0470665769.

CSE 496 (1.5 credits) HUMAN COMPUTER INTERACTION LAB
3 Hours/Week
The laboratory works will be based on theory classes where several programming exercises will need to be performed.

CSE497 (3 credits) NEURAL NETWORKS AND DEEP LEARNING
3 Hours/Week
History and Introduction: History, Basic overview of Neural Networks and Deep Learning and current advances in deep learning, Applied Math and Machine Learning Basics: Linear algebra and basic Calculus, Gradient descent, Logistic regression, Probability and information theory, Bayes' Rule, Numerical computation, Machine Learning basics. Neural Networks Basics: What are neural networks? The linear classifier, The perceptron, activation functions, backpropagation, dropout, Overfitting and underfitting, Gradient-Based Optimization, Feedforward Networks, Shallow neural networks, Deep Neural Networks, Training Neural Networks. Convolutional Networks: The convolution Operation, The Neuroscientific Basis for Convolutional Networks, Overview of Convolutional Neural Networks(CNN), Pooling etc. Recurrent and Recursive Nets: Recurrent Neural Networks(RNN), LSTM, Autoencoders: Undercomplete Autoencoders, Regularized Autoencoders, Stochastic Encoders and Decoders, Denoising Autoencoders etc. Research topics: May include linear factor models, auto encoding, representation learning, Monte Carlo methods, Boltzmann machines, generative models. Applications of Deep Learning: Some applications of deep learning i.e. in Vision applications, NLP application, Speech Recognition etc.

Textbooks

1. Deep Learning. by Ian Goodfellow, YoshuaBengio, Aaron Courville.
2. Pattern Recognition and Machine Learning by Christopher Bishop.
3. The Matrix Cookbook by Kaare Brandt Petersen, Michael Syskind Pedersen.

CSE498 (1.5 credits**)** NEURAL NETWORKS AND DEEP LEARNINGLAB
3 Hours/Week
Student should be able solve the mathematical, statistical and computational challenges of building stable representations for high-dimensional data, such as images, text and audio. Student should be able to use some Deep Learning Softwares like Caffe, Torch, Theano, TensorFlow, Keras, PyTorch, Scikit-Learn etc. Student also learn to solve problem like classification / pattern recognition using deep learning. There will be regular homework and programming assignments based on theory.

Detailed Syllabus
Non-Major Courses
(offrered by other departments)

First Year First Semester

Course Title: History of The Emergence of Independent Bangladesh	**Credits:** 3.0
Course No.: SSS 100D	**Contact hours:** 3 hours/week

Rationale:
This is a special compulsory course for all students of Bachelor program of Shahjalal University of Science and Technology, Sylhet. This course deals with the interrelated themes and topics that are essential to understand the emergence of Bangladesh.

Objectives:

- To give an outline about the concept of liberation war and freedom fighter
- To clarify the role of different people in the liberation war.
- To explain the role of Bangabandhu in liberation war
- To develop an insight about the value of the sacrifice of martyrs for motherland.

Course Contents:

This course deals with the following interrelated themes and topics that are essential to understand the emergence of Bangladesh. These themes include land and people, politics, economy, governance, society, religion and culture, global connections as well as the basic topics on the freedom struggle and War of Liberation. Issues under each of the broad themes will be discussed from the perspective of historical evolution and contemporary significance.

Description of the country and its people: Impact of Geographical features, Ethnic composition of Bangladesh, Development of Bengali Language and its impact, Cultural syncretism and religious tolerance, Distinctive identity of Bangladesh in the context of undivided Bangladesh. **Proposal for undivided sovereign Bengal, the partition of the Subcontinent, 1947 and Foreshadowing Bangladesh:** Rise of communalism under the colonial rule, Lahore Resolution 1940, The proposal of Suhrawardi and Sarat Bose for undivided Bengal: consequences, The creation of Pakistan 1947, Foundation of Awami Muslim League and Foreshadowing Bangladesh. **Pakistan: Structure of the state and disparity;** Central and provincial structure, Influence of Military and Civil bureaucracy, Economic, social and cultural disparity. **Language Movement and quest for Bengali identity:** Misrule by Muslim League and Struggle for democratic politics, The Language Movement: context, phases and International Recognition of Bengali Language, United front of Haque – Vasani – Suhrawardi: election of 1954, consequences. **Military rule: the regimes of Ayub Khan and Yahia Khan (1958-1971):** Definition of military rules and its characteristics, Ayub Khan's rise to power and characteristics of his rule (Political repression, Basic democracy, slamisation), Fall of Ayub Khan and Yahia Khan's rule. **Rise of nationalism and the Movement for self-determination:** Resistance against cultural aggression and resurgence of Bengali culture, Sheikh Mujibur Rahman and the 6 points movement, Reactions: Importance and significance, The Agortola Case 1968. **The mass- upsurge of 1969 and 11 point movement:** Background, Program, Significance. **Election of 1970 and its Impact:** Legal Framework Order (LFO), Programs of different political parties, Election result and centers refusal to comply **Non-cooperation Movement and 7th March Speech, 1971:** The non-cooperation movement, Speech of 7th March: Background of the speech, major characteristics of the speech, impact of this speech, International recognition of 7th March Speech as part of world heritage. **Declaration of Independence of Bangladesh:** Operation Searchlight, Declaration of Independence of Bangladesh by Bangobondhu, Beginning of the Liberation War of Bangladesh. **The war of Liberation 1971:** Genocide, repression of women, refugees, Formation of Bangladesh government and proclamation of Independence, The spontaneous early resistance and subsequent organized resistance (Mukti Fouz, Mukti Bahini, guerillas and the frontal warfare), Publicity Campaign in the war of Liberation (Shadhin Bangla Betar Kendra, the Campaigns abroad and formation of public opinion), Contribution of students, women and the masses (Peoples war) and different political parties, The role of Great powers and the United Nations in the Liberation war, The contribution of India in the Liberation War, The Anti-liberation activities of the occupation army, the Peace Committee, Al-Badar, Al-Shams, Rajakars, pro Pakistan political parties and Pakistani Collaborators , killing of the intellectuals, Trial of Bangabandhu and reaction of the World Community, Formation of joint command and the Victory, The overall contribution of Bangabandhu in the Independence struggle. **The Bangabandhu Regime 1972-1975:** Homecoming; Speech of 10 January, Making of the constitution, Reconstruction of the war-ravaged country, Foreign Policy of Bangabandhu; Bangabandhu's First Speech in the United Nations, The murder of Bangabandhu and his family and the ideological turn-around.

Course Learning Outcomes**:** After the successful completion of the course, the student will be able to-

CLO 1	Know liberation war of Bangladesh and role of freedom fighters
CLO 2	*Know the causes of developing movement and nationalism*
CLO 3	Know different disparities and deprivation of Bangladesh by Pakistan
CLO 4	*Know the declaration and continuing breathtaking moments of liberation war.*
CLO 5	*Know the lifelong contributions of Bangabandhu Sheikh Mujibor Rahman in the creation of independent Bangladesh.*

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/	PLO1	PL02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9

PLO									
CLO1							X		X
CLO2							X		X
CLO3							X		X
CLO4							X		X
CLO5							X		X

Recommended texts:

1. Ahmed, Salahuddin and Bazlul Mobin Chowdhury (eds.), *Bangladesh: National Culture and Heritage: An Introductory Reader* (Dhaka: Independent University Bangladesh, 2004)
2. Harun-or-Roshid, *The Foreshadowing of Bangladesh: Bengal Muslim League and Muslim Politics, 1906-1947* (Dhaka : The University Press Limited, 2012)
3. Jahan Rounaq, *Pakistan: Failure in National Integration,(Dhaka :* The University Press Limited, 1977)
4. Maniruzzaman Talukder, *Radical Politics and the Emergence of Bangladesh, (Dhaka :* Mowla,Brothers, 2003)
5. Muhith, A M A, *History of Bangladesh: A Subcontinental Civilization*, (Dhaka: UPL, 2016)
6. Samad Abdus, *History of Liberation War of Bangladesh*, (Dhaka : Aparajeyo Bangla Prakashani, 2019)
7. Milton Kumar Dev, Md. Abdus Samad, *History of Bangladesh* (Dhaka : Biswabidyalya Prokasoni, 2014)
8. Schendel, Willem van : *A History of Bangladesh* (Cambridge: Cambridge University Press, 2009)
9. †kL gywReyi ingvb : *Amgvß AvZ¥Rxebx, (*XvKv : w` BDwbfvwm©wU †cÖmwjwg‡UW, 2012)
১০. নীহাররঞ্জনরায় : *বাঙালীর ইতিহাস, (*কলকাতা : দে' জ পাবলিশিং, ১৪০২ সাল)
১১. সালাহ্ উদ্দিন আহমেদ ও অন্যান্য (সম্পাদিত), *বাংলাদেশের মুক্তি সংগ্রামের ইতিহাস* ১৯৪৭-১৯৭১, (ঢাকা : আগামী প্রকাশনী, ২০০২)
১২. আবুল মাল আবদুল মুহিত : *বাংলাদেশ: জাতিরাষ্ট্রের উদ্ভব, (ঢাকা :* সাহিত্য প্রকাশ, ২০০০)
১৩. সিরাজুল ইসলাম (সম্পাদিত), *বাংলাদেশের ইতিহাস ১৭০৪-১৯৭১*, ৩ খন্ড, (ঢাকা : এশিয়াটিক সোসাইটি অব বাংলাদেশ, ১৯৯২)
১৪. হারুন-অর-রশিদ : *বঙ্গীয় মুসলিম লীগ পাকিস্তান আন্দোলন বাঙালির রাষ্ট্রভাবনা ও বঙ্গবন্ধু*, (ঢাকা : অন্য প্রকাশন, ২০১৮)
১৫. হাসান হাফিজুর রহমান :বাংলাদেশের স্বাধীনতাযুদ্ধ দলিলপত্র, (সম্পাদিত), (ঢাকা:গণপ্রজাতন্ত্রী বাংলাদেশ সরকার, ১৯৮৫)
১৬. সৈয়দ আনোয়ার হোসেন : *বাংলাদেশের স্বাধীনতাযুদ্ধে পরাশক্তির ভূমিকা*, (ঢাকা :ডানা প্রকাশনী, ১৯৮২)
১৭. মুনতাসীর মামুন ও অন্যান্য, *স্বাধীন বাংলাদেশের অভ্যুদয়ের ইতিহাস*, (ঢাকা: সুবর্ণ, ২০১৭)

১৮. আবু মো দেলোয়ার হোসেন, *স্বাধীন বাংলাদেশের অভ্যুদয়ের ইতিহাস*, (ঢাকা : বিশ্ববিদ্যালয় প্রকাশনী, ২০১৪)
১৯. আশফাক হোসেন, *স্বাধীন বাংলাদেশের অভ্যুদয়ের ইতিহাস*, (ঢাকা: প্রতিশূণ্য প্রকাশন, ২০১৯)
২০. আবু মো দেলোয়ার হোসেন, *বাংলাদেশের ইতিহাস, ১৯০৫-১৯৭১*,
২১. আশফাক হোসেন : *বাংলাদেশের মুক্তিযুদ্ধ ও জাতিসংঘ*, (ঢাকা: বাংলা একাডেমি, ২০০৩)
২২. আবু মো. দেলোয়ার হোসেন, ড. মোহাম্মদ সেলিম (সম্পাদনা) : *বাংলাদেশ ও বহির্বিশ্বে, (*ঢাকা : বাংলাদেশ ইতিহাস সমিতি, ২০১৫)
২৩. আশফাক হোসেন, *বাংলাদেশের মুক্তিযুদ্ধ ও ইন্দ্রিরা গান্ধী* (ঢাকা : সুবর্ণ প্রকাশনী, ২০১৭)

Course Title: Electrical Circuits	**Credits:** 3.0
Course Code: EEE 109D	**Contact hours:** 3 hours/week

Rationale:

This is an introductory course in Electrical and Electronic Engineering, introducing simple electrical DC circuits as well as the technical skills To facilitate necessary knowledge to analyze such simple and complex circuits. It is a course suitable for students pursuing further studies in electrical, electronic or telecommunications engineering as well as some other related engineering disciplines. It gives the through idea about different types of circuit analysis techniques. It also gives a broad idea of single and three phase power systems with various resistive and reactive loads. It contains the relationship between real, apparent and reactive power - including the use of phasor and impedance diagrams, methods of measuring power, calculation of power factor. This course deals with all of this.

Course Objectives:

- To disseminate knowledge about electrical charge, voltage, current and power.
- To give the idea of basic concepts of DC circuit behavior.
- Help the students to conceptualize with the basic theorems of circuit analysis.
- To help the students develop skills to solve mathematical problems of simple and complex electrical circuits.
- To familiarize the students with the basics of AC networks.
- To teach the modeling and analysis of single phase RLC circuits for impedances, voltages, currents, powers and phase shift.

Course Contents:

Circuit variables and elements: Voltage, current, power, energy, independent and dependent sources, and resistance. **Basic laws:** Ohm's law, Kirchhoff's current and voltage laws. **Simple resistive circuits**: Series and parallel circuits, voltage and current division, wye-delta transformation. **Techniques of circuit analysis:** Nodal and mesh analysis including super node and super mesh. **Network theorems:** Source transformation, Thevenin's, Norton's and superposition theorems with applications in circuits having independent and dependent sources, maximum power transfer condition and reciprocity theorem. **Energy storage elements:** Inductors

and capacitors, series parallel combination of inductors and capacitors. **Responses of RL and RC circuits:** Natural and step responses. **Sinusoidal functions:** Instantaneous current, voltage, power, effective current and voltage, average power, phasors and complex quantities, impedance, real and reactive power, power factor. **Analysis of single-phase AC circuits:** Series and parallel RL, RC and RLC circuits, nodal and mesh analysis, application of network theorems in AC circuits.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CLO 1	Explain charge, current, voltage and power, resistance etc.
CLO 2	Interpret the basic circuit laws and circuit analysis techniques.
CLO 3	Design DC circuits and analyze them.
CLO 4	Apply different Network Theorems.
CLO 5	Explain the basics of energy storage elements.
CLO 6	Explain basic AC circuit concepts and responses.
CLO 7	Design RLC circuits and analyze them.
CLO 8	Apply network techniques to AC circuits and networks

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO9
CLO 1	X	X							
CLO 2	X	X							
CLO 3	X	X	X				X	X	
CLO 4	X	X							
CLO 5	X						X		
CLO 6	X	X							
CLO 7	X	X							
CLO 8	X	X					X	X	

Recommended Books

1. Fundamental of Electric Circuits – Charles K. Alexander and Matthew N.O. Sadiku
2. Introductory Circuit Analysis by Robert L.Boylestad.

Course Title: Basic Electrical and Electronic Circuits Lab	**Credits:** 1.5
Course Code: EEE 110D	**Contact hours:** 3 hours/week

Rationale

In this course students will perform experiments to verify practically the theories and concepts learned in EEE-109D. Theoretical knowledge is incomplete without hands on experiments using the basic components and measuring devices used in electrical circuits' analysis. This course teaches the fundamentals of electrical circuits, application of circuit laws, theorems and measuring techniques for DC circuits. It contains broad idea of single-phase power systems with various resistive and reactive loads. It also makes familiar about the relationship between real, apparent and reactive power - including the use of phasor and impedance diagrams, methods of measuring power, calculation of power factor.

Course Objectives

- To provide the students with capability of implementing different real-life dc circuits.
- To provide the students with the techniques of solving of different types of circuits by network theorem.
- To teach the voltage, current and load relationship in a network.
- To facilitate necessary knowledge about transient analysis and steady state analysis of a capacitor and inductor network.
- Helping the students to develop ability in building AC electrical circuits and perform experiments on them.

Course Contents:

In this course students will perform experiments to verify practically the theories and concepts learned in EEE-109.

- To familiarize students with the operation of different electrical instruments.
- To verify the following theorems:
- KCL and KVL theorem,
- Superposition theorem,
- Thevenin's theorem,
- Norton's theorem and
- Maximum power transfer theorem
- RL and RC response.
- Study the frequency response of an RLC circuit and find its resonant frequency.

- Basic electrical element like fan, bulb, calling bell etc connection from 220v AC single phase supply.
- Relevant application based on EEE 109D.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CLO 1	Differentiate different types of electrical instruments and measuring devices.
CLO 2	Experiment different types of circuit analysis theorem and laws
CLO 3	Impart the idea about complex circuit network.
CLO 4	Interpret transient response about capacitor and inductor circuits
CLO 5	Design AC electrical circuits on breadboard and perform measurements with electronic test equipment.
CLO 6	Demonstrate team-based communication skills, magnify their moral standards and apply these in practical life

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO 1	X	X					X	X	
CLO 2	X	X	X						
CLO 3	X	X	X						
CLO 4	X	X	X						
CLO 5	X	X	X				X		
CLO 6						X	X	X	

Recommended Books

1. Fundamental of Electric Circuits – Charles K. Alexander and Matthew N.O. Sadiku
2. Introductory Circuit Analysis by Robert L.Boylestad

Course Title: Effective Communication in English	**Credits:** 2.0

Course Code: ENG 101D	**Contact hours:** 2 hours/week

Rationale:

This course will develop two basic skills i.e. reading and writing. A variety of reading strategies and texts will be used to effectively develop first year students' academic reading skills thereby facilitating their future study. Also, the course focuses on developing the writing skills of students by familiarizing them with grammar rules, providing them with practice thereby enabling them to demonstrate the accurate use of grammar in their writing.

Course Objectives:

- To enable students to write with accuracy
- To facilitate effective and comprehensible writing
- To raise awareness of common errors that occur in writing
- To develop student's ability to understand write-ups on issues of general concern.
- To improve the vocabulary of learners for effective communication

Course Content:

a) Reading

- Different Reading Strategies
- Guessing Meaning from the Context
- Critical Reading (Analyze)
- Critical Reading (Synthesize)
- Critical Reading (Evaluate)
- Annotation
- Summary Writing

Material

- A selection of 08-10 editorials and reports from newspapers/magazines/journals,etc
- Reading texts in New Headway Upper Intermediate Student's Book (Current edition)
- Selected passages from recommended books
- A selection of other material may be supplied as handouts as deemed necessary by the instructor

b) Writing

- Forms and functions of different word categories (Noun, verb, adjective, etc.)
- Aspects and uses of tense
- Subject-verb agreement
- Use of infinitive, gerund, present participle, past participle, modals, causatives, conditionals, subjunctives, modals.
- Use of sentence connectors/ cohesion markers/ punctuation
- Effective combination of sentences (simple, complex, compound)

- Developing a paragraph

Course Learning Outcomes
At the end of the course, students will be able to

CLO 1	Apply grammar rules
CLO 2	Produce grammatically correct meaningful sentences
CLO 3	Express oneself correctly by using appropriate words, phrases, sentences or ideas
CLO 4	Critically reflect on a text (grasp abstract ideas and interpret them effectively, arrive at well reasoned conclusions and solutions).
CLO 5	Extract information from passages accurately

Mapping CLOs to PLOs

CLO \ PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO 1									X
CLO 2									X
CLO 3									X
CLO 4									X
CLO 5									X

Books Recommended

1. Tibbits, E. E. ed. *Exercises in Reading Comprehension*. Longman
2. Liz and John Soars. (Current edition). *New Headway Upper Intermediate Student's Book*.
3. Oxford : Oxford University Press
4. Cliff's TOEFL

Course Title: English Language Lab - I	**Credits:** 1.0
Course Code: ENG 102D	**Contact hours:** 2 hours/week

Course Rationale
This course is designed to improve the speaking and listening skills of students in the English language. Emphasis is laid on proper pronunciation for accurate articulation and recognition of speech sounds as well as correct stress, intonation and language use in varied situations.

Course Objectives
1. To enable students' understanding of the variations in pronunciation
2. To teach proper pronunciation and accurate articulation.
3. To facilitate appropriate stress and intonation in speech.
4. To encourage use of English effectively in everyday situations.
5. To ensure overall improvement of oral communication through listening and speaking.

Course Content
(a) Speaking
- Articulators
- English Phonetic Alphabet (British and American) and International Phonetic Alphabet (IPA)
- Stress rules of English
- Intonation rules and functions of intonation
- Communication Styles and Cultural Context
- Fluency, mistakes, misunderstandings, audience, taboos, self-esteem, confidence
- Activities: dialogue, debate, extempore speech, interview, role-play

(b) Listening
- Basics of listening
- Various types of Pronunciation
- IPA, RP, Transcription
- Different accents and intonation patterns
- Activities for Meaning-focused Listening, Information Transfer Strategies,
- Listening Practice through selection of audio clips.

Course Learning Outcomes
At the end of the course, students will be able to

CLO 1	read the symbols of the International Phonetic Alphabet used to represent the sounds of the English language.
CLO 2	understand all that is being said in English in varied accents
CLO 3	interpret information accurately

CLO 4	apply appropriate intonation and stress patterns in English words and sentences.
CLO 5	produce continuous speech clearly and convincingly.

Mapping CLOs to PLOs

CLO \ PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1									X
CLO2									X
CLO3									X
CLO4									X
CLO5									X

Books Recommended

1. Anderson, A. & Lynch, T. Listening. Oxford: Oxford University Press. 1988
2. Hancock, Mark. English Pronunciation in Use. New York: Cambridge University Press. 2004
3. Anderson, Kenneth, et al. Study Speaking. Cambridge University Press, 2007
4. Hancock, Mark. English Pronunciation in Use. Cambridge University Press, 2004
5. Jones, Daniel. Cambridge English Pronunciation Dictionary. Cambridge University Press, 2011
6. Richards J, et al. Person to Person. Oxford University Press, 2007
7. Richards, Jack C, and David Bohlke. Speak Now: 1. Oxford University Press, 2013
8. Roach, Peter. English Phonetics and Phonology. Cambridge University Press, 2009

Course Title: Coordinate Geometry And Linear Algebra	**Credits:** 3.0
Course Code: MAT101D	**Contact hours:** 3 hours/week

Rationale:

In an increasingly complex world, mathematical thinking, understanding, and skill are more important than ever. MAT 102D will show students how to simplify many types of complex problems using matrix algebra and vector geometry. Students who major in the sciences or engineering are often required to study linear algebra. This course provides a solid foundation for further study in mathematics, the sciences, and engineering.

Course Objectives:

- Engage students in sound mathematical thinking and reasoning
- Provide a setting that prepares students to read and learn mathematics on their own
- Enhance and reinforce the student's understanding of concepts through the use of technology when appropriate

Course Contents:

Coordinate geometry: Coordinate geometry of two dimensions: Change of axes; transformation of co-ordinates; pair of straight lines; general equation of second degree. Coordinate geometry of three dimensions: System of co-ordinates; distances of two points; section formula; projection; direction cosines; equations of planes and straight lines. Matrix: Matrix and matrix operations; different types of matrices; algebraic operations on matrices; cofactors and minors; determinant of a square matrix; adjoint and inverse of a matrix; elementary transformation of matrices; normal and canonical form of a matrix; rank of a matrix; the row-reduced form of a matrix and rank; equivalent systems of linear equations; the general solution of a system of linear equations; homogeneous systems; eigenvalues and eigenvectors; diagonalization of matrices. Vector space: Vector spaces and subspaces; linear dependence and independence; spanning set and basis; coordinates and dimension; null space, row space and column space; change of basis. Linear transformations: Linear transformations; composition of transformations; matrix representation; change of basis; diagonalization representation of a linear transformation by a diagonal matrix; the eigenvalues and eigenvectors of a symmetric matrix; quadratic form; functions of a square matrix. Inner product spaces: Definition and examples; Cauchy-Schwartz inequality; orthogonality; orthonormal basis and Gram-Schmidt process.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CLO 1	Interpret quantitative data verbally, graphically, symbolically and numerically
CLO 2	Determine the size, transpose, inverse, rank, and LU-factorization of a matrix
CLO 3	Perform matrix operations and solve matrix equations
CLO 4	Perform vector operations for vectors in n R
CLO 5	Apply geometric and algebraic properties of vectors in R^n to compute vector additions and scalar multiplications

CLO 6	Use mathematical concepts in problem-solving through integration of new material and modeling

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO 1	X				X				
CLO 2	X				X				
CLO 3	X				X				
CLO 4	X				X				
CLO 5	X				X				
CLO 6	X				X				

Recommended Books

1. Howard Anton and Chris Rorres: Elementary linear algebra applications
2. Steven J. Leon: Linear algebra with applications, Prentice Hall, 1998
3. Rahman and Bhattacharjee: Co-ordinate geometry of two and three dimensions
4. Loney, S. L.: Coordinate Geometry of Two dimensions

First Year Second Semester

Course Title: Electronic Devices & Circuits	**Credits:** 3.0
Course Code: EEE 111D	**Contact hours:** 3 hours/week

Rationale

This course endeavors to build on this knowledge and further expand students' skill in analyzing and designing analogue circuits involving transistors and diodes. The course covers: the basic principles of operation and device characteristics of diodes, Bipolar Junction Transistors (BJT), Junction Filed Effect Transistors (JFET) and Metal Oxide Semiconductor Field Effect Transistors (MOSFET) that underpin the analysis, design and implementation of analogue circuits. Multi-stage amplifiers using BJT and FETs further enhanced the course. Upon completion, students should be able to construct, analyze, verify, and troubleshoot analog circuits using appropriate techniques and test equipment.

Course Objectives:

- To introduce the basic principle operations, device and circuit characteristics of diodes, BJT, JFET, MOSFET and Op-Amp.
- To further develop skill and knowledge in analysis and design of analogue circuits such as amplifiers.
- To introduce the idea about DC and AC analysis of different amplifier circuits.
- To make the students interpret semiconductor theory.

Course Contents:

P-N junction as a circuit element: Intrinsic and extrinsic semiconductors, operational principle of p-n junction diode, contact potential, current-voltage characteristics of a diode, Diode circuits: Half wave and full wave rectifiers, rectifiers with filter capacitor, characteristics of a Zener diode, clamping and clipping circuits. **Bipolar Junction Transistor (BJT) as a circuit element:** current components, BJT characteristics and regions of operation, BJT as an amplifier, biasing the BJT for discrete circuits, small signal equivalent circuit models, BJT as a switch. **Metal Oxide Semiconductor Field Effect Transistor (MOSFET) as circuit element:** structure and physical operation of an enhancement MOSFET, threshold voltage, Body effect, current-voltage characteristics of an enhancement MOSFET, and biasing discrete and integrated MOS amplifier circuits, single-stage MOS amplifiers, MOSFET as a switch, CMOS inverter. **Operational amplifiers (Op-Amp):** Properties of ideal Op-Amps, non-inverting and inverting amplifiers, inverting integrators, differentiator, weighted summer and other applications of Op-Amp circuits. Introduction to photodiode, Laser, Solar cell, Photo detector, LED.

.**Course Learning Outcomes:**

After the successful completion of the course, the student will be able to-

CLO 1	Interpret the basic semiconductor theory
CLO 2	Explain the basis operation of diode, and diode circuits
CLO 3	Design BJT amplifier circuits and perform DC and AC analysis.
CLO 4	Design JFET amplifier circuits and perform DC and AC analysis.
CLO 5	Identify different MOSFET circuits
CLO 6	Formulate an understanding for special purpose MOSFET
CLO 7	Differentiate between switching network of BJT and MOSFETs

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO9
CLO 1	X	X	X	X					
CLO 2	X	X	X	X					

CLO 3	X	X	X				X		
CLO 4	X	X	X				X		
CLO 5	X	X							
CLO 6	X			X			X	X	
CLO 7	X	X		X			X	X	

Recommended Books

1. Electronic Devices and Circuit Theory by Robert L. Boylestad and Louis Nashlesky
2. Microelectronic Circuits- Sedra/Smith
3. Digital logic and Computer Design – M. Morris Mano

Course Title: Electronic Devices & Circuits Lab	**Credits:** 1.5
Course Code: EEE 112D	**Contact hours:** 3 hours/week

Rationale

In this course students will perform experiments to verify practically the theories and concepts learned in EEE-111D. Theoretical knowledge is incomplete without hands on experiments using the basic components and measuring devices. This is an introductory experimental laboratory that explores the design, construction, and debugging of analog electronic circuits. Lectures and two laboratory projects investigate the performance characteristics of diodes, transistors, JFETs, and MOSFETS, including the construction of a small audio amplifier and preamplifier. The course provides opportunity to simulate real-world problems (as given as assignment) and solutions that involve tradeoffs and the use of engineering judgment.

Course objectives:

- Acquaint students with the basic idea about implementing different types diode circuits and investigates the voltage, current relationships.
- To help them develop skills for calculating voltage gain, current gain, overall gain in a multistage BJT, JFET and MOSFET amplifiers.
- To provide the students with capability of implementing different real life analog electronic circuits.

Course Contents:

- To familiarize students with electronics devices and Laboratory Equipment.
- To study of V-l Characteristics curve of P-N junction diode.
- To study of Half-Wave Rectification circuit.
- To study of Full-Wave Rectification circuit (Bridge &Center- tap).
- To study of Clipping and clamping circuit.

- To study MOSFET and BJT characteristics.
- Speech/ Audio amplification using NPN/PNP Transistor.
- MOSFET as an amplifier and switch.
- Different operational amplifier circuits.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CLO 1	Explain operation of diodes.
CLO 2	Design types of diode circuits.
CLO 3	Distinguish and interpret operation of BJT, JFET and MOSFET
CLO 4	Calculating operating point and perform DC analysis.
CLO 5	Differentiate between BJT, JFET and MOSFET amplifier circuits.
CLO 6	Differentiate between BJT and MOSFET switching circuits
CLO 7	Demonstrate team-based communication skills, magnify their moral standards and apply these in practical life

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO9
CLO 1	X	X	X						
CLO 2	X	X	X	X			X		
CLO 3	X	X					X	X	
CLO 4	X	X	X						
CLO 5	X	X							
CLO 6	X	X		X					
CLO 7						X	X		

Recommended Books

1. Fundamental of Electric Circuits – Charles K. Alexander and Matthew N.O. Sadiku
2. Introductory Circuit Analysis by Robert L.Boylestad
3. Electronic Devices and Circuit Theory by Robert L. Boylestad and Louis Nashlesky

4. Microelectronic Circuits- Sedra/Smith

Course Title: Engineering Graphics	**Credit**:1.5
Course No.: IPE 106 D	**Contact hours:** 3 hours/week

Rationale:
This course is a practical application of knowledge pertaining to all geometric features of a whole product, or assembly or sub-assemblies. Pictorial presentation by means of geometric shapes, lines, and dimensions is a must for engineering students. All engineering students need the basic engineering graphics knowledge to express their thoughts and ideas precisely and accurately.

Course Objectives:
- provide the students with necessary skill to read, understand, and create mechanical engineering drawing
- familiarize the students to acquire and use engineering drawing skills on creating accurate, clear sketches of different mechanical objects following the information and instructions
- make students able to draw different types of angle projections, orthographic views, auxiliary, sectional views, isometric views, etc.
- enable students to acquire requisite knowledge required for advanced study of engineering drawing
- apply the drawing and drafting skills as problem-solving tools to resolve the primary design issues

Course Content:
Introduction, Instruments and their uses, First angle and third angle projections, Orthographic drawing, Sectional views. Isometric views, Missing lines and views.

Course Learning Outcomes
Upon successful completion of this course, student have reliably demonstrated the ability to

CLO 1	Understand the basic tools and techniques for making engineering drawings, and apply them to a wide range of engineering fields such as mechanical engineering, civil and architectural design, chemical process, etc.
CLO 2	Create freehand sketches of visual expressions of technical ideas and can interpret common types of engineering drawings
CLO 3	Understand the purpose of geometric shapes, signs and symbols, abbreviations and dimensional values found on engineering drawings
CLO 4	Utilize graphic techniques to understand the relationships between real-world components and views of the components (orthographic, auxiliary, sectional, isometric views, etc.)
CLO 5	Produce a formal engineering drawing according to standard drafting practice
CLO 6	Participate in design reviews or technical meetings in a company concern to mechanical component and assembly suppliers.

Mapping of CLOs with PLOs

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO9
CLO 1	X					X			
CLO 2	X					X			
CLO 3	X					X			
CLO 4	X					X			
CLO 5	X					X			
CLO 6	X					X			

Books Recommended:

1. K.V. Reddy, Textbook of Engineering Drawing, BS Publications, India.
2. K. Rathnam, A First Course in Engineering Drawing, Springer Nature Singapore Pte Ltd.
3. M.B. Shah and B. C. Rana, Engineering Drawing, Dorling Kindersley (India) Pvt Ltd.
4. Colin H. Simmons and Denis E. Maguire, Manual of Engineering Drawing to British and International standards, 2e, Elsevier Newnes, Oxford.
5. K. Morling, Geometric and Engineering Drawing, Elsevier Ltd. USA.

Course Title: Workshop Practice	Credit: 1.0

Course No.: IPE 108 D	**Contact hours:** 2 hours/week

Rationale:
In order to have a balanced overall development of Engineering graduates, it is necessary to integrate theory with practice. Workshop practice has been included in the curriculum to provide hands-on experience using different tools and basic manufacturing practices. By studying Workshop Practice sessional, students will learn to explain the function, use, and application of different working tools, equipment, machine tools, and the technique of manufacturing a product from its raw material. This course also aims to develop the dignity of labor, precision, safety at work, teamwork, and the students' right attitude.

Course Objectives:
- Acquaint students with the hand tools used in practice to fabricate a product;
- Facilitate necessary knowledge about the specification of machine tools used in workshops and manufacturing industries;
- Develop skill in identifying the machine tool components and their respective functions, and performing various machining operations on the machine tools used in practice;
- Help students develop the ability to identify and differentiate the work holding devices used in practice to manufacture a product;
- Encourage the students to provide team effort in product manufacturing.

Course Content:
Introduction to Hand Tools; Study and operation of an Engine Lathe; Study and operation of the Milling Machine; Study and operation of the Bench Drilling Machine; Study and operation of Shaper Machine; and Preparation of a hexagonal nut.

Course Learning Outcomes
After the successful completion of the course, students will be able to:

CLO 1	Sketch, specify and use various work holding tools, marking and measuring tools, cutting tools, finishing tools, and other tools such as hammer, spanner, screwdriver, and wrench, etc.;
CLO 2	Specify various machine tools such as engine lathe, milling machine, bench drilling machine, and shaper machine used in workshops as well as manufacturing industries;
CLO 3	Identify various components of an engine lathe, milling machine, bench drilling machine, and shaper machine and describe their respective functions;
CLO 4	Identify and differentiate the work holding devices used in an engine lathe, milling machine, bench drilling machine, and shaper machine;
CLO 5	Produce a formal engineering drawing according to standard drafting practice
CLO 6	Apply their machining skills to fabricate parts of desired features from a given work piece as per given drawing;
CLO 7	Apply their team effort to make the sequence of operations required for manufacturing a hexagonal nut and prepare it from a given cylindrical work piece as per given drawing.

Mapping of CLOs with PLOs

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO 1	X					X			
CLO 2	X					X			
CLO 3	X					X			
CLO 4	X					X			
CLO 5	X					X			
CLO 6	X					X			
CLO 7	X					X			

Books Recommended:
1. Rajender Singh, Introduction to Basic Manufacturing and Workshop Technology
2. U.K. Singh and Manish Dwivedi, Manufacturing Processes
3. H.N. Gupta, R.C. Gupta, and Arun Mittal, Manufacturing Processes

Course Title: Mechanics, Waves, Heat and Thermodynamics	Credits:3.0
Course Code: PHY 103D	Contact hours: 3 hours/week

Course Objectives
- to learn about Newtonian mechanics and problem-solving technique.
- to know waves behavior and Fourier theorem.
- to learn the basic principles of thermodynamics.

- to understand the radiation.

Course Content

Mechanics: motion in two dimensions, projectile motion, Newton's laws of motion, central forces and gravitation, Kepler's law. **Waves:** simple harmonic motion, damped and forced harmonic vibrations, waves in elastic media. Fourier's theorem and application. **Heat, Thermodynamics and Radiation:** principles of thermometry, zeroth law of thermodynamics. kinetic theory of gases, first and second law of thermodynamics, entropy, blackbody radiation. Wein's law, Planck's law.

Course Learning Outcomes

After the successful completion of the course, students will be able to:

CLO1	explain motion, cause of motion, projectile motion and applying the idea in different cases.
CLO2	apply Kepler's laws to predict path of the Planets and calculate revolving time around the Sun.
CLO3	perform the details calculation about simple harmonic motion for free, damped and forced vibrations.
CLO4	apply laws of thermodynamics.
CLO5	explain the black-body radiation.

Mapping of the CLOs with PLOs

CLO/PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X	X			X				
CLO2	X	X			X				
CLO3	X	X			X				
CLO4	X	X			X				
CLO5	X	X			X				

Recommended Books

1. Halliday, D. and Resnick, R.: Physics (Vol. I and Vol II)
2. Puri, S. P.: Fundamentals of Vibrations and Waves
3. Saha and Srivastava: A Treatise of Heat

Course Title: Calculus	**Credits:** 3.0

Course Code: MAT103D	**Contact hours:** 3 hours/week

Rationale

The focus and themes of the Calculus course address the most important foundations for applications of mathematics in science, engineering and commerce. The course emphasizes the key ideas and historical motivation for calculus, while at the same time striking a balance between theory and application, leading to a mastery of key threshold concepts in foundational mathematics.

Course Objectives

- Establish the fundamental theorems and applications of the calculus of single variable functions
- Help them explore the concepts, properties and aspects of the differential and integral calculus of single variable functions
- Provide students with the mathematical tools necessary for more advanced fields

Course Contents:

Differential Calculus: Functions of a real variables and their plots; limit; continuity and derivatives; physical meaning of derivative of a function; Leibnitz Theorem; Rolle's Theorem; mean value theorem and Taylor's theorem (statement only).Taylor's and Maclaurin's series and expansion of functions; maximum and minimum values of functions; functions of two or three variables; partial and total derivatives. Integral Calculus: Physical meaning of integration of a function; integration as an inverse process of differentiation; different techniques of integrations; definite integrate as the limit of a sum and as an area; definition of Riemann integrals; fundamental theorem of integral calculus and its application to definite integrals; reduction formula; improper integrals; double integration; evaluation of area and volume by integration. Differential Equations: Definition and solution of ordinary differential equation; first order ordinary differential equation; second order ordinary linear differential equation with constant coefficients; initial value problems.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CLO 1	Demonstrate knowledge of basic pre-calculus concepts and skills
CLO 2	Evaluate limits
CLO 3	Recognize continuity and use the properties of continuous functions
CLO 4	Find derivatives of algebraic and trigonometric functions using the definition or basic rules of differentiation

CLO 5	Develop and practice methods of differential calculus with applications
CLO 6	Develop and practice methods of the integral calculus

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO 1	X				X				
CLO 2	X				X				
CLO 3	X				X				
CLO 4	X				X				
CLO 5	X				X				
CLO 6	X				X				

Recommended Books

1. Das and Mukherjee: Differential Calculus
2. Das and Mukherjee: Integral Calculus
3. M.R. Spiegel: Advanced Calculus
4. J. Edwards: Differential Calculus
5. J. Edwards: Integral Calculus
6. R.A. Sardar: Differential Calculus
7. S. L. Ross: Differential equations

Second Year First Semester

Course Title: Cost and Management Accounting	**Credits:** 3.0
Course Code: BUS 243	**Contact hours:** 3 hours/week

Rationale

This subject contributes to the program outcome by expanding students' skill technically through the analysing of different costing method which is preferable for every stakeholder.

Course Objectives

- To acquaint with the cost concepts, cost behavior, and cost accounting techniques that are applied to manufacturing and service businesses.
- To make students capable to interpret cost accounting statements
- To provide the students with the capability to apply theoretical knowledge in decision making
- To help them be able to analyze and evaluate information for cost ascertainment, planning, control of business operations
- To develop to skill to discuss the various techniques available to measure managerial performance and to motivate employees toward organizational goals
- To develop skill to identify and analyze both qualitative and quantitative standards to formulate best control methods

Course Contents:

Introduction to Cost Accounting: Definition of Cost Accounting, Comparison of Cost Accounting and Financial Accounting; The role of Cost Accounting; Methods and Techniques of Cost Accounting; Characteristics of an Ideal Cost Accounting System; **Cost Concepts, Classifications and Statements:** Cost Object; Expenditures, Cost, Expense and Loss; Cost Classifications; Cost Data and Uses; The Chart of Accounts; Statement of Cost of Goods Manufactured and Sold; Cost Statement or Cost Sheet; **Costing and Control of Materials:** Classification of Materials; Accounting for Materials; Store ledger(FIFO & WAM) method; Inventory Planning; Ordering Cost, Holding Cost and EOQ; Effect of Quantity Discounts on EOQ; Safety Stock and Reorder Point; Material Control Methods; Materials Requirement Planning System. Practical problem; **Costing and Control of Labor**: Productivity and Labor Costs; Costs included in Labour; Accounting for Labour; Time Keeping, Computation of total payroll and Allocation of Payroll costs; Different incentive plan; Labour cost Control, Labor Turnover and Control of Labour Turnover; Learning Curve Theory. Practical problem & solution; **Costing and Control of Manufacturing Overhead**: Manufacturing Overhead Costs; Actual Vs. Normal Costing of Manufacturing Overhead; Production Capacity, Predetermined Overhead Rates; Departmental vs. Plant-wise Overhead Rates; Separating Mixed Costs. Scatter-graph; High-low Method and Regression Analysis; Accounting for Manufacturing Overhead; Analysis and Disposition of Under-applied-and Over-applied Overhead; **Contract Costing** : Determination of profit of completed and incomplete contracts; **Introduction of Management Accounting** :Definition-process of Management Accounting, characteristics of Management Accounting, scope of Management Accounting, purpose and objectives of Management Accounting, Comparison of Management Accounting and Financial Accounting; **Cost Terms, Concepts and Classifications:** Cost Behavior (Analysis and Use):General cost classifications- product costs versus period costs- cost classifications on Financial Statements. Types of cost behavior patterns- the Analysis of Mixed Costs, High-low method; **Cost-Volume-Profit Relationships:** The basics of CVP analysis- Break -even analysis- Break-even chart- Sales Mix. Business application and mathematical problem of CVP analysis; **Budget:** Define Budget, Types of Budget, Cash budget, purchase budget, sales budget, flexible budget and Related problems; **Standard Costing:** Meaning and Objectives- Types of ratios. Standard Costing and its uses for making business decision. Variance calculation, Decision making process from these calculations.

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CLO 1	Discuss about how cost accounting is used for decision making and performance evaluation
CLO 2	Competent to demonstrate how materials, labor and overhead costs are added to a product at each stage of the production cycle
CLO 3	Express the place and role of cost accounting in the modern economic environment
CLO 4	Recognize and apply the skills necessary for carrying out effective management decision-making and strategic management planning
CLO 5	Select the costs according to their impact on business and society
CLO 6	Interpret the impact of the selected costs method
CLO 7	Design management control process in different business areas

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO 1	X						X		X
CLO 2	X						X		X
CLO 3	X						X		X
CLO 4	X						X		X
CLO 5	X						X		X
CLO 6	X						X		X
CLO 7	X						X		X

Recommended Books:

1. Cost Accounting –Volume-1 by Basu and Das;
2. Managerial Accounting by Ray H. Garrison, Eric W. Noreen

Course Title: Basic Physics Lab	**Credits:** 1.5
Course Code: PHY 202D	**Contact hours:** 3 hours/week

Rationale

In this course students will perform some laboratory experiments that will help to visualize some fundamental concepts of physics.

Course Objectives:

- To enable the student to carry out some fundamental experiments for finding out the numerical values of some physical parameters based on various laws, principle and theorem of physics.

Course Content:

Mechanics:

1. Determination of moment of inertia of a flywheel
2. Determination of "g" by and moment of inertia of a compound pendulum

Properties of matter:

1. Determination of Young's Modulus by the method of bending
2. Determination of Rigidity Modulus by Static method
3. Using a flat spiral spring: a) Verification of Hooke's Law and determination of stiffness constant; b) Determination of "g" and the effective mass of the spring; c) Determination of modulus of rigidity of the material of the spring.

Electricity:

1. Determination of galvanometer resistance by half deflection method

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CLO 1	Exploit conservation of mechanical energy to analyze system undergoing both rotation and translation for finding the moment of inertia of uniformly shaped, rigid body
CLO 2	Interpret load vs. strain graphs and perform calculations using elastic module
CLO 3	Describe the elastic deformations of materials needed in our daily life
CLO 4	Determine the value of acceleration due to gravity "g" by using a compound pendulum from the measured time period and compare the predicted result with the published data
CLO 5	Determine the resistance of a galvanometer by half deflection method. Understand the various electrical components used in the experiment and construct circuits based on the circuit diagram
CLO 6	Calculate the % and standard error using the error analysis rules

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO9
CLO 1	X								
CLO 2	X	X		X					
CLO 3	X	X							
CLO 4	X	X		X					
CLO 5	X	X	X						
CLO 6	X			X					

Recommended Books

1. Worsnop, B.L. and Flint, H.T. :Advanced Practical Physics
2. Chowdhury, S. A. and Basak, A. K. :Byaboharik PadarthaBidya
3. Ahmed, G. and Uddin, M.S. :Practical Physics

Course Title: Basic Statistics & Probability	**Credits:** 3.0
Course Code: STA 202D	**Contact hours:** 3 hours/week

Rationale:

Acquiring knowledge on the statistical tools and techniques for exploring and analyzing the data.

Course Objectives

- Provide the knowledge on fundamental concepts of statistical methods
- Acquaint students with the basic tools of exploratory data analysis,
- Facilitate necessary knowledge about bivariate data analysis
- Make students understand the basic concepts of probability and probability distribution,
- Help the students conceptualize basic theories in Stochastic processes including Markov chain and queuing theory

ontent of Course

Frequency distribution of data: population and sample. Collection and representation of statistical data. Tabulation of data. Class intervals. Frequency distribution, discrete, continuous and cumulative distributions. Histograms and frequency polygons. Graphical representation of data. Statistical measures: measures of central tendency - arithmetic mean, median, mode, geometric mean, weighted average, harmonic mean. Measures of dispersion - range, standard deviation, variance, coefficient of variation, moments, skewness, kurtosis. Correlation theory: linear correlation. Measures of correlation and its significance. Regression and curve fitting: linear and non-linear regression. Methods of least squares. Curve fitting. Probability: definition of probability and related concepts. Laws of probability. Discrete and continuous random variables. Mathematical expectations. Conditional probability. Probability distributions: binomial, Poisson and normal distributions and their properties. Stochastic process. Markov chain (discrete and continuous). Queuing theory – birth and death process in queuing. Examples from computer science. Queuing models – elementary concepts.

Course Learning Outcomes

After the successful completion of the course, the student will be able to-

CLO 1	Explain basic concepts of statistics and describe various statistical tools
CLO 2	Construct frequency distribution and present data graphically
CLO 3	Compute and interpret different measures of central tendency, location, dispersion, and shape characteristics
CLO 4	Analyze bivariate data and interpret the results
CLO 5	Calculate probability of an event and derive probability distribution of a random variable
CLO 6	Explain stochastic processes and apply Markov chain and queuing theory

Mapping CLOs to PLOs

CLO/ PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X			X					
CLO2	X			X					
CLO3	X			X					
CLO4	X			X					
CLO5	X			X					
CLO6	X			X					

Text Books

1. Devore J., (2009), Probability and Statistics for Engineering and the Sciences, 8th Edition, Brooks/Cole, Cengage Learning, California

2. Montgomery, D.C., Runger, G.C., (2003), Applied Statistics and Probability for Engineers, 3rd Edition, John Wiley & Sons, Inc., NY
3. Ross, S.M., (2007), Introduction to Probability Models, 9th Edition, Academic Press, NY

Reference Books:

1. Barlow R J, Statistics: A Guide to the Use of Statistical Methods in the Physical Sciences, Wiley, NY
2. Chisholm J S R & Morris R M, Mathematical Methods in Physics, North Holland
3. Hoel P G, Elementary Statistics, 3rd Ed, John Wiley, NY
4. Loveday R, Practical Statistics and Probability, Cambridge University Press, London
5. Mostafa M G, Methods of Statistics, Bangladesh

Second Year Second Semester

Course Title: Digital Logic Design	**Credits:** 3.0
Course Code: EEE 201D	**Contact hours:** 3 hours/week

Rationale

The main aim of this course is to provide sound knowledge of the principles and practices of digital systems, both at the device and circuit level. The course covers topics in digital electronics including: Number Theory, Boolean Algebra, Logic Circuits, Logic Minimization Techniques, Multiplexers, Adders, Flip-Flops, Counters, Registers, State Machines, Memory Circuits, Digital / Analog Conversion, Programmable Logic Circuits and Microcomputer Bus Architecture. Upon completion, students should be able to construct, analyze, verify, and troubleshoot digital circuits using appropriate techniques and test equipment.

Course Objectives:

- To make students understand the fundamental principles in design and implementation of digital logic circuits including combinational circuits, sequential circuits, and finite state machines.
- To develop skills to perform decimal, octal, hexadecimal, and binary conversions.
- To provide the knowledge to apply Boolean algebra to solve logic functions.
- To help students in learning the analysis of pulse circuits.
- To help students in learning the analysis of digital multiplexing circuits.
- To help students in learning the analysis of logic family interfaces.
- To help students in learning the analysis of logic switching circuits.

Course Contents:

Logic Families: TTL, CMOS, ECL, Tristate
Logic Gates: AND, OR, NAND, NOR, X-OR, X-NOR, Circuit Design
Flip flops: SR, JK, D, Master Slave, Application, and Synchronization
Logic Circuits: Coder, Decoder, Mux, Dmux
Counters: Synchronous, Asynchronous, Up/Down, Ripple, Cascading
Registers: Shift registers
Memory Devices: ROM, RAM, Static, Dynamic, Memory Operation
Arithmetic Circuits: Adder, Carry, Look Ahead, ALU
PAL: Micro-program Control, FPGA, HDLA

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CLO 1	Perform simple arithmetic in binary, octal, hexadecimal, BCD number systems
CLO 2	Manipulate logic expressions using binary Boolean algebra
CLO 3	Generate the prime implicates of logic functions of 5 or fewer variables using graphical (Karnaugh map) method, and to obtain their minimal two-level implementations with and without don't cares.
CLO 4	Analyze combinational circuits
CLO 5	Use basic functional & timing (clocking) properties of latches & flip-flops.
CLO 6	Analyze synchronous sequential circuits to extract next state/output functions
CLO 7	Translate a word statement specifying the desired behavior of a simple sequential system into a finite state machine (FSM), to simplify and build the architecture that consists of state register and next state/output logic
CLO 8	Implement simple digital systems using controller and basic data path components such as registers, memories, counters, multiplexers, ALUs, etc.

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO 1	X	X		X					

CLO 2	X	X	X	X					
CLO 3	X	X	X	X	X				
CLO 4	X	X	X	X	X				
CLO 5	X	X	X	X	X				
CLO 6	X	X	X	X					
CLO 7	X	X	X	X			X		
CLO 8	X	X		X			X		

Recommended Books

1. Digital Logic Design by Morris Mano
2. Digital Systems by Ronald Tocci, Neal Widmer, Greg Moss
3. Digital Principles and Applications by Donald P Lech, Albert Paul Malvino and Goutam Saha

Course Title: Digital Logic Design Lab	**Credits:** 1.5
Course Code: EEE 202D	**Contact hours:** 3 hours/week

Rationale

The main aim of this course is to provide practical knowledge of the principles and practices of digital systems, both at the device and circuit level. The course covers practical experiments of the topics of digital electronics including: Number Theory, Boolean Algebra, Logic Circuits, Logic Minimization Techniques, Multiplexers, Adders, Flip-Flops, Counters, Registers, State Machines, Memory Circuits, Digital / Analog Conversion, Programmable Logic Circuits and Microcomputer Bus Architecture. Upon completion, students should be able to construct, analyze, verify, and troubleshoot digital circuits using appropriate techniques and test equipment.

Course Objectives

- Help students to conceptualize the fundamental principles in design and implementation of digital logic circuits including combinational circuits, sequential circuits, and finite state machines.
- To develop skills to perform decimal, octal, hexadecimal, and binary conversions.
- To provide the knowledge to apply Boolean algebra to solve logic functions.
- To help students in learning the analysis of pulse circuits.
- To help students in learning the analysis of digital multiplexing circuits.
- To help students in learning the analysis of logic family interfaces.

Course Contents:

- Logic circuits using combination of gates
- To construct and study the following logic gates: AND, OR, NOT. NAND, NOR, EXOR
- Verify the DE Morgan's Law: Law(I) and Law (II)
- To verify different kind of applications of Boolean algebra.
- To construct an AND gate by diode resistors and observe its characteristics.
- To verify the characteristics of Exclusive OR and Exclusive NOR using basic logic gate.
- Verification of De-Morgan's Theorem for 2 input Variable.
- To simplify the given Boolean function by using K-map and implement it with logic Diagram.
- ABCD to 7 Segment Decoder
- Study of 4-bit BCD adder.
- Study of Asynchronous & Synchronous R-S Flip-Flop.
- Study of J-K Flip-Flop.
- Study of 4-bit binary Ripple Counter.
- Verilog HDL Basics.
- Project with PAL/FPGA/Microcontroller

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CLO 1	Manipulate logic expressions using binary Boolean algebra
CLO 2	Generate the prime implicates of logic functions of 5 or fewer variables using graphical (Karnaugh map) method, and to obtain their minimal two-level implementations with and without don't cares.
CLO 3	Understand and analyze combinational circuits
CLO 4	Use basic functional & timing (clocking) properties of latches & flip-flops.
CLO 5	Analyze synchronous sequential circuits to extract next state/output functions
CLO 6	Translate a word statement specifying the desired behavior of a simple sequential system into a finite state machine (FSM), to simplify and build the architecture that consists of state register and next state/output logic

CLO 7	Implementing simple digital systems using controller and basic data path components such as registers, memories, counters, multiplexers, ALUs, etc.
CLO 8	Demonstrate team-based communication skills, magnify their moral standards and apply these in practical life

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO /CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO 1	X	X	X	X					
CLO 2	X	X	X	X	X				
CLO 3	X	X	X	X					
CLO 4	X	X	X	X	X				
CLO 5	X	X	X	X					
CLO 6	X	X	X	X			X		
CLO 7	X	X	X				X		
CLO 8						X	X		

Recommended Books

1. Digital Logic Design by Morris Mano
2. Digital Systems by Ronald Tocci, Neal Widmer, Greg Moss
3. Digital Principles and Applications by Donald P Lech, Albert Paul Malvino and Goutam Saha

Course Title: Complex Variables, Laplace Transform And Fourier Series	**Credits:** 3.0
Course Code: MAT 204D	**Contact hours:** 3 hours/week

Rationale

The course gives the students a sound knowledge of Fourier transforms along with Fourier integrals, Laplace Transformation and complex variables.

Course Objectives

- To facilitate with basic ideas about complex variables
- To provide knowledge on Laplace transformation
- To acquaint with Fourier series and analysis

Course Contents:

Complex Variables: Complex numbers and their properties; De Moivre's theorem and its application; locus problem; functions of a complex variable; limit and continuity of a function of complex variable; analytical functions; the Cauchy-Riemann equations; Cauchy's theorem; singularity and poles; residues; simple contour integration and their uses in solving boundary value problems. Laplace Transformations: Definition of Laplace transform; Laplace transform of different functions; first shift theorem; inverse Laplace transform; linearity property; use of first shift theorem and partial functions; Laplace transform of derivatives; Laplace transform of an integral; the Heaviside unit function; the unit impulse function; the second shift theorem; periodic functions; convolutions; solution of ordinary differential equations by Laplace transform. Fourier Series: Fourier series; convergence of Fourier series; Fourier analysis; Fourier transforms

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CLO 1	Define the complex number system, complex functions and integrals of complex functions
CLO 2	Apply the results/theorems in complex analysis to complex valued functions
CLO 3	Understand Laplace transforms
CLO 4	Discuss about analytic function and how to check analyticity based on Cauchy – Riemann equation
CLO 5	Compute Fourier and Laplace transforms
CLO 6	Represent periodic functions using Fourier series

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO9
CLO 1	X				X				
CLO 2	X				X				
CLO 3	X				X				
CLO 4	X				X				
CLO 5	X				X				

CLO 6	**X**				**X**				

Recommended Books

1. KK Kodaira: Introduction to Complex analysis
2. H Jaffreys and B Jaffreys: Methods of Mathematical Physics
3. Spiegel, M. R.: Laplace Transform
4. Khanna, M. L.: Laplace Transforms

Course Title: Principles of Economics	**Credits:** 3.0
Course Code: ECO105D	**Contact hours:** 3 hours/week

Rationale

This ECO 105 course provides an introduction to the main ideas and concepts involved in modern economics and attempts to provide students with an understanding of how the economy works, what type of problems economists attempt to solve, and how they set about trying to solve them. The course is primarily concerned with the analysis of individual decision-making agents, the behaviour of firms and industries in the economy (microeconomics), on the economy as a whole (macroeconomics) and the inherent problems facing underdeveloped and developing countries (economic development).

Course Objectives

- To provide a brief and simple introduction to the subject matter and scope of Economics.
- To provide a brief and simple introduction to the subject matter and scope of Macroeconomics.
- To provide students with an understanding of economic theories and analysis in the field of development economics.

Course Contents:

Introduction to Microeconomics: Definition and scope; basic concepts and tools—PPF and circular flow model; fundamental economic problems and solution systems; Concepts of demand, supply and equilibrium; Concepts of elasticity, different types of elasticities, their applications; Concepts of total and marginal utility; Concepts of production, cost and profit, characteristics of different types of markets. **Introduction to Macroeconomics**: Key macroeconomic indicators and their performance measurement - GNP, GDP, inflation, unemployment; money, functions of money, function of commercial and central bank, monetary policy; fiscal policy and structure of govt. budget. **Development and related issues:** Growth and development; concept of poverty and poverty measures; HDI; key human-socio-economic development indicators of Bangladesh, Sustainable Development Goals (SDG).

Course Learning Outcomes:

After the successful completion of the course, the student will be able to-

CLO 1	Understand the analysis of individual decision-making agents, the behaviour of firms and industries in the economy
CLO 2	Understand the concept of elasticity quantitatively and qualitatively in economic analysis and know differences between different types of markets
CLO 3	Explain macroeconomic concepts and use simple economic models to interpret the behaviour of key macroeconomic variables
CLO 4	Understand monetary and fiscal policy and Government budget
CLO 5	Understand the main issues confronting underdeveloped and developing countries

Mapping of Course Learning Outcomes to Program Learning Outcomes

PLO/ CLO	PLO 1	PLO 2	PLO 3	PLO 4	PLO 5	PLO 6	PLO 7	PLO 8	PLO 9
CLO 1	X				X				
CLO 2	X				X				
CLO 3	X				X				
CLO 4	X				X				
CLO 5	X				X				
CLO 6	X				X				

Recommended Books

1. Arnold, R. A. (2014): Economics, South Western Publishing Company, Eleventh Edition
2. Bangladesh Economic Review relevant issues.
3. Mankiw, N. G. (2012): Principles of Economics, Thomson South Western Publishing, Sixth Edition
4. Samuelson, P. A. and Nordhaus,W. D. (2009): Economics, McGraw-Hill USA, Nineteenth Edition.
5. Todaro, M. P. and Smith,S. C. (2012): Economics of Development in the Third World, Longman, Eleventh Edition

List of Non-Major Courses
(offered for other departments)

Course No.	Credit	Dept	Yr-Sem	Course Title
CSE 103M	2	MAT	1-2	Introduction to Computer Language
CSE 104M	2	MAT	1-2	Introduction to Computer Language Lab
CSE103	2	*	1-1/1-2	Introduction to Computer Language
CSE104	2	*	1-1/1-2	Introduction to Computer Language Lab
CSE 203	2	*	2-1/2-2	Introduction to Computer Language
CSE 204	2	*	2-1/2-2	Introduction to Computer Language Lab
CSE113	2	*	1-1/1-2	Introduction to Programming with Python
CSE114	2	*	1-1/1-2	Introduction to Programming with Python Lab
CSE213	2	*	2-1/2-2	Introduction to Programming with Python
CSE214	2	*	2-1/2-2	Introduction to Programming with Python Lab
CSE313	2	*	3-1/3-2	Introduction to Programming with Python
CSE314	2	*	3-1/3-2	Introduction to Programming with Python Lab
CSE216	3	*	2-1/2-2	Python Programming Lab
CSE316	3	*	3-1/3-2	Python Programming Lab
CSE102	3	*	1-1/1-2	Introduction to Computing Applications
CSE202	3	*	2-1/2-2	Introduction to Computing Applications
CSE302	3	*	3-1/3-2	Introduction to Computing Applications
CSE205	2	*	2-1/2-2	Database Management and Programming
CSE206	3	*	2-1/2-2	Database Management and Programming Lab
CSE207M	3	MAT	2-1	Algorithm Design and Analysis
CSE208M	1.5	MAT	2-1	Algorithm Design and Analysis Lab
CSE211M	3	MAT	2-2	Data Structure
CSE212M	1.5	MAT	2-2	Data Structure Lab
CSE 219W	3.0	SWE	2-1	Computer Architecture
CSE 317W	3.0	SWE	3-1	Computer Networking
CSE 318W	1.5	SWE	3-1	Computer Networking Lab

* (A-ARC, B-CEP, C-CEE, D-CSE, E-EEE, F-FET, G-IPE, H-PME, I-BMB, J-GEB, K-CHE, L-GEE, M-MAT, N-PHY, O-STA, P-FES, Q-MEE, W-SWE, a-ANP, b-BNG, c-ECO, d-ENG, e-PSS, f-PAD, g-SCW, h- SOC, i-BUS)

Detailed Syllabus
Non-Major Courses
(offered for other departments)

Course Title: Introduction to Computing Application	**Credits:** 3.0
Course Code: CSE 102*	**Contact hours:** 6 hours/week

Rationale

This is a dedicated lab course that aims to make students familiar with the basic computing applications that are frequently used in real life. The main focus of this course is to introduce the familiar applications (i.e. Microsoft Excel, Microsoft Access, SPSS) of Spreadsheet analysis, Database, and Statistical analysis. It covers preparing and working with spreadsheets, formatting spreadsheets, creating charts, using tools, and printing worksheets. It also covers the basics of databases, table

designs, indexing, creating queries, creating forms, and printing reports. As part of statistical analysis, it also includes the introduction of statistical analysis, operation commands, data definition and manipulation commands, and procedure commands. All of these will help students to have a good understanding and hands-on experience with the above mentioned basic applications.

Course Objectives

- To assist students in developing a good understanding of the vastly used basic computing applications.
- To help students understand the basic concepts of spreadsheets, databases, and statistical analysis.
- To facilitate the basic knowledge about the workflows of the applications like Microsoft Excel/Access, SPSS.
- To assist students in developing their hands-on skills on the mentioned applications.
- To make students capable of applying their knowledge on the mentioned areas in real life.

Course Contents:

Spreadsheet Analysis: Introduction (Spreadsheet & its Applications, Menus & Toolbars), Working with Spreadsheets (Converting files to different formats, Importing, Exporting, Spreadsheet addressing, Computing data, Mathematical operations, Using formulas), Formatting Spreadsheets (Border & shading, Highlighting values, Visibility, Sorting, Filtering, Validation, Consolidation, Subtotal), Creating Charts (Selecting charts, Formatting charts, label, scaling, etc.), Using Tools (Error Checking, Spell Checks, Macros), Printing worksheet. **Database Applications:** Introduction (Database concepts, Tables, Queries, Forms, Reports), Working with Databases (Creating Tables, Table Design, Indexing, Entering data, importing data), Creating Queries (SQL statements, Setting relationship, using wizards), Creating Forms, Creating & printing reports. **Statistical Analysis:** Introduction, Operation commands, Data definition, manipulation commands, and procedure commands like LIST, DESCRIPTIVES, FREQUENCIES, CROSSTABS, T-TEST, ANOVA, REGRESS, etc.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Explain the basic concepts of a spreadsheet.
CLO 2	Prepare a working spreadsheet with all necessary functionalities.
CLO 3	Use spreadsheets to smartly store and manipulate large data.
CLO 4	Explain the concepts of database (tables, queries, mapping, indexing, forms, etc.)
CLO 5	Design databases.
CLO 6	Create a working database with tables and relations, and make queries in the database.
CLO 7	Describe the basic concepts of statistical analysis.
CLO 8	Analyze (calculate mean/mode/median, regression, correlation, interpolation, etc.) a large amount of statistical data and generate analysis reports using applications like SPSS.

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Course Title: Introduction to Computer Language	**Credits:** 2.0
Course No.: CSE 103*	**Contact hours:** 2 hours/week

Rationale:

To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow the students to write their programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:

- To provide students a basic understanding of computer hardware and how a computer works
- To make students understand the basic terminology used in computer programming
- To facilitate with knowledge of how to write, compile and debug programs in the C language
- To help students write programs involving decision structures, loops, functions, and pointers
- To help students develop skills on standard programming practices and how to build up their logic and how to implement them.

Course Contents:

Computer Basics: Concept on Computer Hardware, Software, and its classification. **C-Language:** Preliminaries, Program constructs variables and data types in C. Input and output. Character and formatted I/O; Arithmetic Expressions and Assignment statements; Loops and Nested loops; Decision making; Arrays, Functions; Arguments and local variables, Calling Functions and arrays. Recursion and Recursive functions; Structures within a structure. Files; File functions for sequential and Random I/O. Pointers; Pointers and structures; Pointer and functions;

Pointer and arrays; Operation and Pointer; Pointer and memory addresses; Operations on Bits; Bit Operation; Bit field; Advanced features; Standard and library.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand the concepts of computer hardware and how it works
CLO 2	Recall the basic terminology used in computer programming
CLO 3	Construct, compile and debug programs in the C language
CLO 4	Apply control-flow tools such as loop, if-else, etc.
CLO 5	Understand the usage of pointers, structures, and some advanced topics
CLO 6	Employ standard programming practices

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook
1. Schaum's Outline of Programming with C by Byron S. Gottfried
2. C: The Complete Reference by Herbert Schildt

Course Title: Introduction to Computer Language Lab	**Credits:** 2.0
Course No.: CSE 104*	**Contact hours:** 4 hours/week

Rationale:
To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow the students to write their programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:
- To help to develop skills to work with C compilers and how to use run programs on the computer
- To foster the analytical and critical knowledge to build up logic and implement them using programming language C
- To facilitate necessary knowledge about how to design programs involving decision structures, loops, functions, and pointers
- To help to develop skills to debug codes by giving an in-depth idea about different syntax errors, exceptions and how to fix them

Course Contents:
Computer Basics: Concept on Computer Hardware, Software, and its classification. **C-Language:** Preliminaries, Program constructs variables and data types in C. Input and output. Character and formatted I/O; Arithmetic Expressions and Assignment statements; Loops and Nested loops; Decision making; Arrays, Functions; Arguments and local variables, Calling Functions and arrays. Recursion and Recursive functions; Structures within a structure. Files; File functions for sequential and Random I/O. Pointers; Pointers and structures; Pointer and functions; Pointer and arrays; Operation and Pointer; Pointer and memory addresses; Operations on Bits; Bit Operation; Bit field; Advanced features; Standard and library. **Problem Solving:** Basic Calculator, Odd/Even Test, Showing Letter Grade from Number, Drawing different shapes using Asterisks (*), GCD, Palindrome, Fibonacci Series, Geometric Mean, Quadratic Formula, Cumulative Sum, Cumulative Product, Weighted Average, Generating Prime Number using Sieve of Eratosthenes, Prime Factorization, Big Mod, SOD, NOD, Permutation, Combination, Finding Areas, Basic Geometry Problems, Factorial, Leap Year, Tower of Hanoi, String Manipulation: Vowel and Consonant Count, Reversing a Word, Matrix Multiplication, Piglatin Generator.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Recognize C compilers and necessary tools to run programs on the computer
CLO 2	Interpret logic and implement them using C
CLO 3	Design programs involving decision structures, loops, functions, and pointers
CLO 4	Debug codes by using the in-depth idea about different types of errors and exceptions
CLO 5	Implement knowledge of programming to solve real-life problems

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook
1. Schaum's Outline of Programming with C by Byron S. Gottfried
2. C: The Complete Reference by Herbert Schildt

Course Title: Introduction To Computer Language	**Credits:** 2.0

Course No.: CSE 103M	**Contact hours:** 2 hours/week

Rationale:
To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow the students to write their own programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:
- To provide students a basic understanding of computer hardware and how a computer works
- To make students understand the basic terminology used in computer programming
- To facilitate with knowledge of how to write, compile and debug programs in the C language
- To help students write programs involving decision structures, loops, functions, and pointers
- To help students develop skills on standard programming practices and how to build up their own logic and how to implement them.

Course Contents:
Computer Basics: Concept on Computer Hardware, Software and its classification, Compiler vs Interpreter. **Programming Language:** Basic concept, Overview of programming languages, Problem Solving Techniques, and Data Flow Diagram. **C-Language:** Preliminaries, Program constructs, variables, and data types in C. Input and output. Character and formatted I/O; Variables and Constants; Precedence and Associativity; Arithmetic Expressions and Assignment statements; Loops and Nested loops; Decision making; Arrays, Functions; Arguments and local variables, Calling Functions and arrays. Recursion and Recursive functions; Structures within a structure. Files; File functions for sequential and Random I/O. Pointers; Pointers, union, and structures; Pointer and functions; Pointer and arrays; Operation and Pointer; Pointer and memory addresses; Operations on Bits; Bit Operation; Bit field; Advanced features; Standard and library. **Sorting:** Insertion sort, selection sort, bubble sort, merge sort. **Searching:** Linear search, binary search, application of Binary Search- finding elements in a sorted array, finding nth root of a real number, solving equations.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand the concepts of computer hardware and how it works
CLO 2	Recall the basic terminology used in computer programming
CLO 3	Construct, compile and debug programs in the C language
CLO 4	Apply control-flow tools such as loop, if-else, etc.
CLO 5	Understand the usage of pointers, structures, and some advanced topics
CLO 6	Apply basic searching and sorting algorithms
CLO 7	*Employ standard programming practices*

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department
Textbook
1. Schaum's Outline of Programming with C by Byron S. Gottfried
2. C: The Complete Reference by Herbert Schildt

Course Title: Introduction to Computer Language Lab	**Credits:** 2.0
Course No.: CSE 104M	**Contact hours:** 4 hours/week

Rationale:
To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow the students to write their own programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:
- To help to develop skills to work with C compilers and how to use run programs on the computer
- To foster the analytical and critical knowledge to build up logic and implement them using programming language C
- To facilitate necessary knowledge about how to design programs involving decision structures, loops, functions, and pointers
- To help to develop skills to debug codes by giving an in-depth idea about different syntax errors, exceptions and how to fix them
- To provide the knowledge of using basic searching and sorting algorithms

Course Contents:
Introduction: Any kind of formatted input and output, Header Files, Using Different Flags, Using Online Judges, Command Line Parameters. **Variables:** Local, Global, Static, Garbage Value, Escape Sequence. **Library Functions:** Mathematical Functions like abs, fabs, ceil, floor, exp, cos, sin, tan, acos, asin, atan, cosh, sinh, tanh, pow, long, fmod, sqrt, rand, srand; I/O Functions like scanf, printf, sscanf, fscanf, sprintf, fprintf, putchar, getchar, puts, gets, fgets, feputs; Character Manipulation Functions like tolower, toupper, toascii; String Manipulation Function like strlen,

strcmp, strdup, strcpy etc. **Structures:** Creating Custom Data Type, Operator Overloading. **File:** Opening File in Different Mode, Closing, Creating, Processing and Closing a File, Casting Standard Input Output to a File. **Sorting:** Insertion sort, selection sort, bubble sort, merge sort, Sorting based on two or more criteria. **Searching:** Linear search, binary Search, application of inary Search- finding an element in a sorted array, finding nth root of a real number, solving equations. **Problem Solving:** Basic Calculator, Odd/Even Test, Showing Letter Grade from Number, Drawing different shapes using Asterisks (*), GCD, Palindrome, Fibonacci Series, Geometric Mean, Quadratic Formula, Cumulative Sum, Cumulative Product, Weighted Average, Generating Prime Number using Sieve of Eratosthenes, Prime Factorization, Big Mod, SOD, NOD, Permutation, Combination, Finding Areas, Basic Geometry Problems, Factorial, Leap Year, Tower of Hanoi, String Manipulation: Vowel and Consonant Count, Reversing a Word, Matrix Multiplication, Piglatin Generator, Encoding and Decoding Cypher Text, Basic Stack and Queue, Storing Students' Personal Information and Manipulating and Answering Queries based on it, Singly Linked List, Doubly Linked List.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Recognize C compilers and necessary tools to run programs on the computer
CLO 2	Interpret logic and implement them using C
CLO 3	Design programs involving decision structures, loops, functions, and pointers
CLO 4	Debug codes by using the in-depth idea about different types of errors and exceptions
CLO 5	Implement knowledge of programming to solve real-life problems
CLO 6	Implement basic searching and sorting algorithms

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook
1. Schaum's Outline of Programming with C by Byron S. Gottfried
2. C: The Complete Reference by Herbert Schildt

Course Title: Introduction to Programming With Python	**Credits:** 2.0
Course Code: CSE113*	**Contact hours:** 2 hours/week

Rationale
In this current world, most of the research works require computational data analysis of corresponding fields. This requirement has emphasized the necessity of a knowledge of computer programming for all the researchers. For research-related purposes, computer programming using Python is one of the best choices. This course is designed with the purpose to make students acquainted with programming using python and make them comfortable to deal with computational data analysis.

Course Objectives
- Help them conceptualize basic theories of computer programming
- Make the students understand fundamental components of python programming
- To develop skills for writing computer programs using all necessary branches of Python
- Accumulate basic ideas about data structures and data manipulations

Course Contents:
Computer Basics: Concept on Computer Hardware, Software and its classification, Compiler vs Interpreter. **Using the Python Interpreter:** Invoking the Interpreter, Argument Passing, Interactive Mode, The Interpreter, and Its Environment, Source Code Encoding; **An Informal Introduction to Python:** Using Python as a Calculator- Numbers, Strings, Lists. First Steps Towards Programming; **More Control Flow Tools:** if Statements, for Statements, The range() Function, break and continue Statements, and else Clauses on Loops, pass Statements, Defining Functions; **More on Defining Functions:** Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations, **Intermezzo:** Coding Style; **Data Structures:** More on Lists- Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, The del statement, Tuples and Sequences, Sets, Dictionaries, Looping Techniques, More on Conditions, . Comparing Sequences and Other Types; **Modules:** More on Modules- Executing modules as scripts, The Module Search Path, Compiled" Python files, Standard Modules, The dir() Function, Packages- Importing * From a Package, Intra-package References, Packages in Multiple Directories, matplotlib, numpy, other common necessary packages; **Input and Output:** Fancier Output Formatting, Old string formatting, **Reading and Writing Files**: Methods of File Objects, Saving structured data with JSON; **Errors and Exceptions:** Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions. **Classes:** A Word About Names and Objects, Python Scopes and Namespaces, Scopes and Namespaces Example, A First Look at Classes, Class Definition Syntax, Class Objects, Instance Objects, Method Objects, Class and Instance Variables, Random Remarks, Inheritance, Multiple Inheritance, Private Variables, Odds and Ends, Iterators, Generators, Generator Expressions.
Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Implement knowledge of Python for writing computer programs
CLO 2	Design solutions of real-life problems using necessary components of Python
CLO 3	Identify errors from a program and use exception handlers to handle errors and exceptions
CLO 4	Implement Object Oriented Programming and modular concepts
CLO 5	Design basic data structures to solve efficient data storage issues
CLO 6	Apply knowledge of programming in data analysis and manipulation

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook:
1. Learning Python, By Mark Lutz, 5th Edition
2. Think Python, By Allen B. Downey
3. The Python Tutorial, Official documentation of Python

Course Title: Introduction to Programming With Python	**Credits:** 2.0
Course Code: CSE114*	**Contact hours:** 4 hours/week

Rationale
In this current world, most of the research works require computational data analysis of corresponding fields. This requirement has emphasized the necessity of a knowledge of computer programming for all the researchers. For research-related purposes, computer programming using Python is one of the best choices. This course is designed with the purpose to make students acquainted with programming using python and make them comfortable to deal with computational data analysis.

Course Objectives
- Help them conceptualize basic theories of computer programming
- Make the students understand fundamental components of python programming
- To develop skills for writing computer programs using all necessary branches of Python
- Accumulate basic ideas about data structures and data manipulations

Course Contents:

Laboratory works based on theory classes and basic problem solving from rosalind.info using Pycharm, Jupyter, and Anaconda IDEs.
Computer Basics: Concept on Computer Hardware, Software and its classification, Compiler vs Interpreter. **Using the Python Interpreter:** Invoking the Interpreter, Argument Passing, Interactive Mode, The Interpreter, and Its Environment, Source Code Encoding; **An Informal Introduction to Python:** Using Python as a Calculator- Numbers, Strings, Lists. First Steps Towards Programming; **More Control Flow Tools:** if Statements, for Statements, The range() Function, break and continue Statements, and else Clauses on Loops, pass Statements, Defining Functions; **More on Defining Functions:** Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations, **Intermezzo:** Coding Style; **Data Structures:** More on Lists- Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, The del statement, Tuples and Sequences, Sets, Dictionaries, Looping Techniques, More on Conditions. Comparing Sequences and Other Types; **Modules:** More on Modules- Executing modules as scripts, The Module Search Path, Compiled" Python files, Standard Modules, The dir() Function, Packages- Importing * From a Package, Intra-package References, Packages in Multiple Directories, matplotlib, numpy, other common necessary packages; **Input and Output:** Fancier Output Formatting, Old string formatting, **Reading and Writing Files**: Methods of File Objects, Saving structured data with JSON; **Errors and Exceptions:** Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions. **Classes:** A Word About Names and Objects, Python Scopes and Namespaces, Scopes and Namespaces Example, A First Look at Classes, Class Definition Syntax, Class Objects, Instance Objects, Method Objects, Class and Instance Variables, Random Remarks, Inheritance, Multiple Inheritance, Private Variables, Odds and Ends, Iterators, Generators, Generator Expressions.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Implement knowledge of Python for writing computer programs
CLO 2	Design solutions of real-life problems using necessary components of Python
CLO 3	Identify errors from a program and use exception handlers to handle errors and exceptions
CLO 4	Implement Object Oriented Programming and modular concepts
CLO 5	Design basic data structures to solve efficient data storage issues
CLO 6	Apply knowledge of programming in data analysis and manipulation

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook:
1. Learning Python, By Mark Lutz, 5th Edition
2. Think Python, By Allen B. Downey
3. The Python Tutorial, Official documentation of Python

Course Title: Python Programming Lab	**Credit:** 3 credits
Course Code: CSE 116	**Contact hours:** 6 Hours/Week

Rationale:
In this current world, most of the research works require computational data analysis of corresponding fields. This requirement has emphasized the necessity of a knowledge of computer programming for all the researchers. For research-related purposes, computer programming using Python is one of the best choices. This course is designed with the purpose to make students acquainted with programming using python and make them comfortable to deal with computational data analysis.

Objectives:
- Help them conceptualize basic theories of computer programming
- Make the students understand fundamental components of python programming
- To develop skills for writing computer programs using all necessary branches of Python
- Accumulate basic ideas about data structures and data manipulations
- To help them to apply the knowledge of programming for data storage, manipulation, and presentation

Course Contents:
Computer Basics: Concept on Computer Hardware, Software and its classification, Compiler vs Interpreter.
Using the Python Interpreter: Invoking the Interpreter, Argument Passing, Interactive Mode, The Interpreter, and Its Environment, Source Code Encoding; **An Informal Introduction to Python:** Using Python as a Calculator- Numbers, Strings, Lists. First Steps Towards Programming; **More Control Flow Tools:** if Statements, for Statements, The range() Function, break and continue Statements, and else Clauses on Loops, pass Statements, Defining Functions; **More on Defining Functions:** Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations, **Intermezzo:** Coding Style; **Data Structures:** More on Lists- Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, The del statement, Tuples and Sequences, Sets, Dictionaries, Looping Techniques, More on Conditions, Comparing Sequences and Other Types; **Modules:** More on Modules- Executing modules as scripts, The Module Search Path, Compiled" Python files, Standard Modules, The dir() Function,

Packages- Importing * From a Package, Intra-package References, Packages in Multiple Directories, **Input and Output:** Fancier Output Formatting, Old string formatting, **Reading and Writing Files**: Methods of File Objects, Saving structured data with JSON; **Errors and Exceptions:** Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions. **Classes:** A Word About Names and Objects, Python Scopes and Namespaces, Scopes and Namespaces Example, A First Look at Classes, Class Definition Syntax, Class Objects, Instance Objects, Method Objects, Class and Instance Variables, Random Remarks, Inheritance, Multiple Inheritance, Private Variables, Odds and Ends, Iterators, Generators, Generator Expressions. **Python Numpy:** Numpy intro, creating arrays, array indexing, array slicing, data types, array shape, array iterating, array join, array split, array search, array sort, array filter, random, ufunc. **Python Pandas:** Pandas series, DataFrames, Read CSV, Read JSON, Analyzing Data, Correlations, Plotting. **Python Matplotlib:** Intro, Pyplot, Markers, Line, Subplots, Scatter, Bars, Histograms, Pie Charts.

Course Learning Outcome: After the successful completion of this course students will be able to

CLO1	Implement knowledge of Python for writing computer programs
CLO2	Design solutions of real-life problems using necessary components of Python
CLO3	Identify errors from a program and use exception handlers to handle errors and exceptions
CLO4	Implement Object Oriented Programming and modular concepts
CLO5	Design basic data structures to solve efficient data storage issues
CLO6	Apply knowledge of programming in data analysis and manipulation
CLO7	Apply knowledge of programming for different graphical data representation techniques

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook:
1. Learning Python, By Mark Lutz, 5th Edition
2. Think Python, By Allen B. Downey
3. The Python Tutorial, Official documentation of Python

Course Title: Introduction to Computing Application	**Credits:** 3.0
Course Code: CSE 202*	**Contact hours:** 6 hours/week

Rationale

This is a dedicated lab course that aims to make students familiar with the basic computing applications that are frequently used in real life. The main focus of this course is to introduce the familiar applications (i.e. Microsoft Excel, Microsoft Access, SPSS) of Spreadsheet analysis, Database, and Statistical analysis. It covers preparing and working with spreadsheets, formatting spreadsheets, creating charts, using tools, and printing worksheets. It also covers the basics of databases, table designs, indexing, creating queries, creating forms, and printing reports. As part of statistical analysis, it also includes the introduction of statistical analysis, operation commands, data definition and manipulation commands, and procedure commands. All of these will help students to have a good understanding and hands-on experience with the above mentioned basic applications.

Course Objectives

- To assist students in developing a good understanding of the vastly used basic computing applications.
- To help students understand the basic concepts of spreadsheets, databases, and statistical analysis.
- To facilitate the basic knowledge about the workflows of the applications like Microsoft Excel/Access, SPSS.
- To assist students in developing their hands-on skills on the mentioned applications.
- To make students capable of applying their knowledge on the mentioned areas in real life.

Course Contents:

Spreadsheet Analysis: Introduction (Spreadsheet & its Applications, Menus & Toolbars), Working with Spreadsheets (Converting files to different formats, Importing, Exporting, Spreadsheet addressing, Computing data, Mathematical operations, Using formulas), Formatting Spreadsheets (Border & shading, Highlighting values, Visibility, Sorting, Filtering, Validation, Consolidation, Subtotal), Creating Charts (Selecting charts, Formatting charts, label, scaling, etc.), Using Tools (Error Checking, Spell Checks, Macros), Printing worksheet. **Database Applications:** Introduction (Database concepts, Tables, Queries, Forms, Reports), Working with Databases (Creating Tables, Table Design, Indexing, Entering data, importing data), Creating Queries (SQL statements, Setting relationship, using wizards), Creating Forms, Creating & printing reports. **Statistical Analysis:** Introduction, Operation commands, Data definition, manipulation commands, and procedure commands like LIST, DESCRIPTIVES, FREQUENCIES, CROSSTABS, T-TEST, ANOVA, REGRESS, etc.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Explain the basic concepts of a spreadsheet.
CLO 2	Prepare a working spreadsheet with all necessary functionalities.
CLO 3	Use spreadsheets to smartly store and manipulate large data.
CLO 4	Explain the concepts of Database (tables, queries, mapping, indexing, forms, etc.)
CLO 5	Design databases.
CLO 6	Create a working database with tables and relations, and make queries in the database.
CLO 7	Describe the basic concepts of statistical analysis.
CLO 8	Analyze (calculate mean/mode/median, regression, correlation, interpolation, etc.) a large amount of statistical data and generate analysis reports using applications like SPSS.

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PLO of the corresponding department

Course Title: Introduction to Computer Language	**Credits:** 2.0
Course No.: CSE 203*	**Contact hours:** 2 hours/week

Rationale:

To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow the students to write their own programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:

- To provide students a basic understanding of computer hardware and how a computer works
- To make students understand the basic terminology used in computer programming
- To facilitate with knowledge of how to write, compile and debug programs in the C language
- To help students write programs involving decision structures, loops, functions, and pointers

- To help students develop skills on standard programming practices and how to build up their own logic and how to implement them.

Course Contents:
Computer Basics: Concept on Computer Hardware, Software, and its classification. **C-Language:** Preliminaries, Program constructs variables and data types in C. Input and output. Character and formatted I/O; Arithmetic Expressions and Assignment statements; Loops and Nested loops; Decision making; Arrays, Functions; Arguments and local variables, Calling Functions and arrays. Recursion and Recursive functions; Structures within a structure. Files; File functions for sequential and Random I/O. Pointers; Pointers and structures; Pointer and functions; Pointer and arrays; Operation and Pointer; Pointer and memory addresses; Operations on Bits; Bit Operation; Bit field; Advanced features; Standard and library.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Understand the concepts of computer hardware and how it works
CLO 2	Recall the basic terminology used in computer programming
CLO 3	Construct, compile and debug programs in the C language
CLO 4	Apply control-flow tools such as loop, if-else, etc.
CLO 5	Understand the usage of pointers, structures, and some advanced topics
CLO 6	Employ standard programming practices

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook
1. Schaum's Outline of Programming with C by Byron S. Gottfried
2. C: The Complete Reference by Herbert Schildt

Course Title: Introduction to Computer Language Lab	**Credits:** 2.0
Course No.: CSE 204*	**Contact hours:** 4 hours/week

Rationale:
To familiarize the student with basic concepts of computer programming and developer tools. To present the syntax and semantics of the "C" language as well as data types offered by the language. To allow the students to write their own programs using standard language infrastructure regardless of the hardware or software platform.

Objectives:
- To help to develop skills to work with C compilers and how to use run programs on the computer
- To foster the analytical and critical knowledge to build up logic and implement them using programming language C
- To facilitate necessary knowledge about how to design programs involving decision structures, loops, functions, and pointers
- To help to develop skills to debug codes by giving an in-depth idea about different syntax errors, exceptions and how to fix them

Course Contents:
Computer Basics: Concept on Computer Hardware, Software, and its classification. **C-Language:** Preliminaries, Program constructs variables and data types in C. Input and output. Character and formatted I/O; Arithmetic Expressions and Assignment statements; Loops and Nested loops; Decision making; Arrays, Functions; Arguments and local variables, Calling Functions and arrays. Recursion and Recursive functions; Structures within a structure. Files; File functions for sequential and Random I/O. Pointers; Pointers and structures; Pointer and functions; Pointer and arrays; Operation and Pointer; Pointer and memory addresses; Operations on Bits; Bit Operation; Bit field; Advanced features; Standard and library. **Problem Solving:** Basic Calculator, Odd/Even Test, Showing Letter Grade from Number, Drawing different shapes using Asterisks (*), GCD, Palindrome, Fibonacci Series, Geometric Mean, Quadratic Formula, Cumulative Sum, Cumulative Product, Weighted Average, Generating Prime Number using Sieve of Eratosthenes, Prime Factorization, Big Mod, SOD, NOD, Permutation, Combination, Finding Areas, Basic Geometry Problems, Factorial, Leap Year, Tower of Hanoi, String Manipulation: Vowel and Consonant Count, Reversing a Word, Matrix Multiplication, Piglatin Generator.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Recognize C compilers and necessary tools to run programs on the computer
CLO 2	Interpret logic and implement them using C
CLO 3	Design programs involving decision structures, loops, functions, and pointers
CLO 4	Debug codes by using the in-depth idea about different types of errors and exceptions
CLO 5	Implement knowledge of programming to solve real-life problems

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook
1. Schaum's Outline of Programming with C by Byron S. Gottfried
2. C: The Complete Reference by Herbert Schildt

Course Title: Database Management and Programming	**Credits:** 2.0
Course No.: CSE 205*	**Contact hours:** 2 hours/week

Rationale:
To familiarize the student with basic concepts of database management systems and their programming.

Objectives:
- To provide knowledge on different issues involved in the design and implementation of a database system.
- To facilitate with knowledge of physical and logical database designs, database modeling, relational, hierarchical and network models
- To provide basic idea on data manipulation language to query, update, and manage a database
- To help to develop an understanding of essential DBMS concepts such as database security, integrity, concurrency, distributed database, Client/Server (Database Server), Data Warehousing.
- To provide knowledge to design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Course Contents:
Computer Basics: Concept on Computer Hardware, Software and its classification, networking, and Internet**. Introduction to Database:** Database Management System, Relational Database management System, Entity-Relationship Model, Relational Model, SQL, Sorting, Indexing, Integrity Constraints, Transaction Concept, Database System Architecture. **Database Management:** Creating a Database, Opening a Database, Modifying a Database, Modifying a Database Structure, Indexing, Sorting, Searching a Database, Designing a Customer Screen, Designing a Report, Designing a Menu. **Database Programming:** Programming concept, A Simple Program, Memory variables, Constants, Operators, Commands, Arrays, Macros, Different Type of Processing, Procedures, Functions. Programming for Data Entries, Update, Report, Menu and Searching.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Describe the fundamentals of Database systems.
CLO 2	Differentiate between different database models.
CLO 3	Design a Relational Database.
CLO 4	Create queries in relational databases.
CLO 5	Use indexing for databases.
CLO 6	Implement concurrency control mechanisms.

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook
1. Database System Concepts – Abraham Silberschratz, Henry K. Korth, S. Sudarshan.
2. Fundamentals of Database Systems - Benjamin/Cummings.

Course Title: Database Management and Programming Lab	**Credits:** 3.0
Course No.: CSE 206*	**Contact hours:** 6 hours/week

Rationale:
Database System LAB course will concentrate on the design and implementation of a database system and applying SQL query.

Objectives:
- To help them understand different issues involved in the design and implementation of a database system.
- To facilitate knowledge about physical and logical database designs, database modeling, relational, hierarchical and network models.
- To help them understand and use data manipulation language to query, update, and manage a database
- To develop skills to design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Course Contents:
Computer Basics: Students will learn the basic concepts of Windows operating system, Word Processor software, SpreadSheet software, and Presentation software. **Database Management**: Students will learn to create, modify, indexing, sorting a database, Designing a customer Screen, designing a report, designing a menu.

Database Programming: Students will learn SQL, Basic structure of SQL Queries, Query-by-example, Nested sub queries, Complex queries, Integrity constraints, Authorization.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Apply functions to process numeric data
CLO 2	Use SQL query to grouping data.
CLO 3	Apply DDL to insert, update and manage database schema
CLO 4	Design a Relational Database
CLO 5	Use indexing for databases
CLO6	Use Nested sub queries, complex queries
CLO7	Maintain integrity and consistency of data

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook

1. Database System Concepts – Abraham Silberschratz, Henry K. Korth, S. Sudarshan
2. Fundamentals of Database Systems - Benjamin/Cummings.

Course Title: Algorithm Design and Analysis	**Credits:** 3.0
Course No.: CSE 207M	**Contact hours:** 3 hours/week

Rationale:
To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:

- To familiarize with the asymptotic performance of algorithms
- To familiarize with rigorous correctness proofs for algorithms
- To demonstrate familiarity with major algorithms and data structures
- To facilitate with the necessary knowledge about important algorithmic design paradigms and methods of analysis
- To develop skills to synthesize efficient algorithms in common engineering design situations

Course Contents:
Analysis of Algorithm: Asymptotic analysis: Recurrences, Substitution method, Recurrence tree method, Master method, Divide and Conquer Algorithms. **Hash Table:** Hash tables, hash function, open addressing, perfect hashing, single and multi-probe hashing. **Greedy Algorithms:** Elements and properties of Greedy algorithms, fractional knapsack, Huffman Coding. **Dynamic Programming:** Elements of DP (Optimal substructure, overlapping subproblem), Coin change-related problem, 0-1 knapsack, Longest Common Subsequence finding problems, LCS and LIS/LDS variations, Matrix Chain Multiplication. Red Black Tree and Binomial Heaps, Stassen's algorithm. **Graph**: Strongly Connected Components, Euler Path, Articulation Point, Bridge, Bi-connected Components, Dijkstra's Shortest Path Algorithm, Bellman-Ford algorithm, and negative cycle detection, Floyd-Warshall all pair shortest path algorithm, shortest path in Directed Acyclic Graph. **Minimum Spanning Tree (MST):** Prim's algorithm and Kruskal's algorithm. **Network Flow:** Flow Networks, Max-Flow Min-Cut Theorem, Ford Fulkerson method and its limitation, Edmonds Karp algorithm, Maximum Bipartite Matching, minimum path cover, minimum edge cover. **Matrix Operation:** Matrix Chain Multiplication. **Backtracking/Branch-and-Bound:** Permutation, Combination, 8-queen problem, 15-puzzle problem, Graph Coloring, N-queen problem, Hamiltonian cycle, Branch and Bound in backtracking. Traveling salesman problem. **Geometric algorithm:** Properties of Line-Segment intersection, Convex-hull, Closest pair problem. **Number Theory:** Chinese Remainder Theorem, Euler phi, extended Euclid, application of prime factorization application of phi. RSA public key generation**,** NP-Completeness, NP-hard and NP-complete problems. **String Matching Algorithms:** Naïve string matching algorithm, Rabin Karp algorithm, String matching with finite automata, Knuth Morris Pratt (KMP) algorithm, Trie, Suffix tree, and Suffix Array. Basic combinatorics, Probability, and Game theory. Least Common Ancestor, Range Minimum Query, Polynomials, DFT, and FFT.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Analyze the computational and memory complexities of algorithms
CLO 2	Prove the correctness of algorithms
CLO 3	Explain how and why the algorithms work
CLO 4	Apply the algorithms to solve real-life problems
CLO 5	*Decide when to use which algorithm*
CLO 6	Synthesize algorithms to design complex solutions
CLO 7	Explain why one algorithm works better than others in different scenarios

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department.

Textbook

1. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson.
2. 102 Combinatorial Problems byTituAndreescu&Zuming Feng
3. Problem-Solving Methods in Combinatoricsby Pablo Soberón
4. Algorithms by Robert Sedgewick and Kevin Wayne.

Course Title: Algorithm Design and Analysis Lab	**Credits:** 1.5
Course No.: CSE 208M	**Contact hours:** 3 hours/week

Rationale:
To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:
- To familiarize with the asymptotic performance of algorithms
- To familiarize with rigorous correctness proofs for algorithms
- To demonstrate a familiarity with major algorithms and data structures
- To facilitate with necessary knowledge about important algorithmic design paradigms and methods of analysis
- To develop skills to synthesize efficient algorithms in common engineering design situations

Course Contents:
Hash Table: Hash tables, hash function, open addressing, perfect hashing, single and multi-probe hashing. **Greedy Algorithms:** Elements and properties of Greedy algorithms, fractional knapsack, Huffman Coding. **Dynamic Programming:** Elements of DP (Optimal substructure, overlapping sub problem), Coin change related problem, 0-1 knapsack, Longest Common Subsequence finding problem, LCS and LIS/LDS variations, Matrix Chain Multiplication. Red black Tree and Binomial Heaps, Stassen's algorithm. **Graph**: Strongly Connected Components, Euler Path, Articulation Point, Bridge, Bi-connected Components, Dijkstra's Shortest Path Algorithm, Bellman –Ford algorithm and negative cycle detection, Floyd-Warshall all pair shortest path algorithm, shortest path in Directed Acyclic Graph. **Minimum Spanning Tree (MST):** Prim's algorithm and Kruskal's algorithm. **Network Flow:** Flow Networks, Max-Flow Min-Cut Theorem, Ford Fulkerson method and its limitation, Edmonds Karp algorithm, Maximum Bipartite Matching, minimum path cover, minimum edge cover. **Matrix Operation:** Matrix Chain Multiplication. **Backtracking/Branch-and-Bound:** Permutation, Combination, 8-queen problem, 15-puzzle problem, Graph Coloring, N-queen problem, Hamiltonian cycle, Branch and Bound in backtracking. Traveling salesman problem. **Geometric algorithm:** Properties of Line-Segment intersection, Convex-hull, Closest pair problem. **Number Theory:** Chinese Remainder Theorem, Euler phi, extended

Euclid, application of prime factorization application of phi. **RSA public key generation,** NP Completeness, NP hard and NP complete problems. **String Matching Algorithms:** Naïve string matching algorithm, Rabin Karp algorithm, String matching with finite automata, Knuth Morris Pratt (KMP) algorithm, Trie, Suffix tree and Suffix Array. Basic combinatorics, Probability and Game theory. Least Common Ancestor, Range Minimum Query, Polynomials, DFT and FFT.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Implement all the listed algorithms
CLO 2	Construct bug free and efficient codes for the algorithms
CLO 3	Construct efficient solution of complex problems using suitable algorithms
CLO 4	Analyze the computational and memory complexities of algorithms
CLO 5	Prove the correctness of algorithms
CLO 6	Explain how and why the algorithms work
CLO 7	Apply the algorithms to solve real life problems
CLO 8	*Decide when to use which algorithm*
CLO 9	Synthesize algorithms to design complex solutions

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PLO of corresponding department.

Textbook
1. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson.
2. 102 Combinatorial Problems by Titu Andreescu & Zuming Feng
3. Problem-Solving Methods in Combinatorics by Pablo Soberón
4. Algorithms by Robert Sedgewick and Kevin Wayne

Course Title: Data Structures	**Credits:** 3.0
Course No.: CSE 211M	**Contact hours:** 3 hours/week

Rationale:
To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter

and showing the correctness of algorithms and studying their computational complexities.

Objectives:

- To explain the purpose and mathematical background of algorithm analysis
- To facilitate necessary knowledge about the abstract data types of stacks, queues, and deques
- To familiarize with a variety of ways that linearly and weakly ordered data can be stored, accessed, and manipulated
- To facilitate necessary knowledge about the characteristics and optimal behavior of hash tables for access and retrieval
- To provide the knowledge of various sorting algorithms and the run-time analysis required to determine their efficiencies
- To help them understand various tree traversal techniques and graph algorithms

Course Contents:

Internal Data Representation: Specification, representation. **Asymptotic analysis:** Recurrences. **Stack and Queue:** Basic stack operations (push/pop/peek), stack-class implementation using Array and linked list, in-fix to post-fix expressions conversion and evaluation, balancing parentheses using stack, basic queue operations (enqueue, dequeue), circular queue/ dequeue, queue-class implementation using array and linked list, application- Josephus problem, Palindrome checker using stack and queue. **Sorting:** merge sort, quick sort (randomized quicksort), and distribution sort (counting sort, radix sort, bucket sort), lower bounds for sorting. **Linked List:** Singly Linked List, Doubly Link List, Traversal, Insertion, Deletion, and Modification. **Searching:** Application of Binary Search- finding elements in a sorted array, finding nth root of a real number, solving equations. **Heap:** Min-heap, max-heap, Fibonacci-heap, applications-priority queue, heap sort. **Set Operations & Disjoint Set:** Union finding, path compression. **Binary Tree (BIT):** Binary tree representation using array and pointers, traversal of Binary Tree (in-order, pre-order, and post-order). Binary Index tree, Segment tree, Range Minimum Query (RMQ), Improvement of BIT: AVL Tree. **Graph:** Graph representation (adjacency matrix/adjacency list), basic operations on graph (node/edge insertion and deletion), traversing a graph: Breadth-first search (BFS), Depth-first search (DFS), Topological Sort.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Interpret the examples of relationships between data.
CLO 2	Analyze computational and memory complexities of algorithms that are used to manipulate data using standard data structures.
CLO 3	Apply stacks/queues/deques to store and extract sequential data.
CLO 4	Apply recursions to divide a problem and conquer the solution.
CLO 5	Decide when to use which data structure.
CLO 6	Differentiate between graphs and trees and make graphs/trees when these are best suited for manipulating the data.
CLO 7	Demonstrate efficiency in inserting data into a data structure and searching/retrieving data from a data structure.
CLO8	Design data structures to store and manipulate data while solving real-life problems.

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook

1. Theory and Problems of Data Structures *(Schaum's Outline Series)* – Seymour Lipschutz.
2. Data Structures and Algorithms: Annotated Reference with Examples *(1st Edition)* – G. Barnett, Luca D. Tongo.

Course Title: Data Structure Lab	**Credits:** 1.5
Course No.: CSE 212M	**Contact hours:** 3 hours/week

Rationale:

To provide the students with solid foundations in the basic concepts of programming: data structures and algorithms. To teach the students how to select and design data structures and algorithms that are appropriate for problems that they might encounter and showing the correctness of algorithms and studying their computational complexities.

Objectives:

- To explain the purpose and mathematical background of algorithm analysis
- To facilitate necessary knowledge about the abstract data types of stacks, queues, and deques
- To familiarize with a variety of ways that linearly and weakly ordered data can be stored, accessed, and manipulated
- To facilitate necessary knowledge about the characteristics and optimal behavior of hash tables for access and retrieval
- To provide the knowledge of various sorting algorithms and the run-time analysis required to determine their efficiencies
- To help them understand various tree traversal techniques and graph algorithms

Course Contents:

Creation and Manipulation of linear data structures: linked list, stacks, and queues. **Creation and Manipulation of non-linear data structures:** B-trees and heaps, disjoint set. **Implementing** sorting, searching and hashing techniques, string processing. **Stack and Queue:** Basic stack operations (push/pop/peek), stack-class implementation using Array and linked list, in-fix to post-fix expressions conversion and evaluation, balancing parentheses using stack, basic queue operations (enqueue, dequeue), circular queue/ dequeue, queue-class implementation using array and linked list, application- Josephus problem, Palindrome checker using stack and queue. **Sorting:** merge sort, quick sort (randomized quicksort), and distribution sort (counting sort, radix sort, bucket sort), lower bounds for sorting. **Linked List:** Singly Linked List, Doubly Link List, Traversal, Insertion, Deletion, and Modification. **Searching:** Application of Binary Search- finding elements in a sorted array, finding nth root of a real number, solving equations. **Heap:** Min-heap, max-heap, Fibonacci-heap, applications-priority queue, heap sort. **Set Operations & Disjoint Set:** Union finds, path compression. **Binary Tree (BIT):** Binary tree representation using array and pointers, traversal of Binary Tree (in-order, pre-order, and post-order). Binary Index tree, Segment tree, Range Minimum Query (RMQ), Improvement of BIT: AVL Tree. **Graph:** Graph representation (adjacency matrix/adjacency list), basic operations on graph (node/edge insertion and deletion), traversing a graph: Breadth-first search (BFS), Depth-first search (DFS), Topological Sort**.**

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Interpret the examples of relationships between data.
CLO 2	Analyze computational and memory complexities of algorithms that are used to manipulate data using standard data structures.
CLO 3	Apply stacks/queues/deques to store and extract sequential data.
CLO 4	Apply recursions to divide a problem and conquer the solution.
CLO 5	Decide when to use which data structure.
CLO 6	Differentiate between graphs and trees and make graphs/trees when these are best suited for manipulating the data.
CLO 7	Demonstrate efficiency in inserting data into a data structure and searching/retrieving data from a data structure.
CLO8	Design data structures to store and manipulate data while solving real-life problems.

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook

1. Advanced Data Structures, Peter Brass
2. Data Structures – Seymour Lipschutz, Schaum's Outlines Series.
3. Introduction to Algorithms Thomas H. Cormen, Charles E. Leiserson

Course Title: Introduction to Programming With Python	**Credits:** 2.0
Course Code: CSE 213*	**Contact hours:** 2 hours/week

Rationale

In this current world, most of the research works require computational data analysis of corresponding fields. This requirement has emphasized the necessity of a knowledge of computer programming for all the researchers. For research-related purposes, computer programming using Python is one of the best choices. This course is designed with the purpose to make students acquainted with programming using python and make them comfortable to deal with computational data analysis.

Course Objectives

- Help them conceptualize basic theories of computer programming
- Make the students understand fundamental components of python programming
- To develop skills for writing computer programs using all necessary branches of Python
- Accumulate basic ideas about data structures and data manipulations

Course Contents:

Computer Basics: Concept on Computer Hardware, Software and its classification, Compiler vs Interpreter. **Using the Python Interpreter:** Invoking the Interpreter, Argument Passing, Interactive Mode, The Interpreter, and Its Environment, Source Code Encoding; **An Informal Introduction to Python:** Using Python as a Calculator- Numbers, Strings, Lists. First Steps Towards Programming; **More Control Flow Tools:** if Statements, for Statements, The range() Function, break and continue Statements, and else Clauses on Loops, pass Statements, Defining Functions; **More on Defining Functions:** Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations, **Intermezzo:** Coding Style; **Data Structures:** More on Lists- Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, The del statement, Tuples and Sequences, Sets, Dictionaries, Looping Techniques, More on Conditions, Comparing Sequences and Other Types; **Modules:** More on Modules- Executing modules as scripts, The Module Search Path, Compiled" Python files, Standard Modules, The dir() Function, Packages- Importing * From a Package, Intra-package References, Packages in Multiple Directories, matplotlib, numpy, other common necessary packages; **Input and Output:** Fancier Output Formatting, Old string formatting, **Reading and Writing Files**: Methods of File Objects, Saving structured data with json; **Errors and Exceptions:** Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions. **Classes:** A Word About Names and Objects, Python

Scopes and Namespaces, Scopes and Namespaces Example, A First Look at Classes, Class Definition Syntax, Class Objects, Instance Objects, Method Objects, Class and Instance Variables, Random Remarks, Inheritance, Multiple Inheritance, Private Variables, Odds and Ends, Iterators, Generators, Generator Expressions.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Implement knowledge of Python for writing computer programs
CLO 2	Design solutions of real-life problems using necessary components of Python
CLO 3	Identify errors from a program and use exception handlers to handle errors and exceptions
CLO 4	Implement Object Oriented Programming and modular concepts
CLO 5	Design basic data structures to solve efficient data storage issues
CLO 6	Apply knowledge of programming in data analysis and manipulation

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook:
1. Learning Python, By Mark Lutz, 5th Edition
2. Think Python, By Allen B. Downey
3. The Python Tutorial, Official documentation of Python

Course Title: Introduction to Programming With Python	**Credits:** 2.0
Course Code: CSE 214*	**Contact hours:** 4 hours/week

Rationale
In this current world, most of the research works require computational data analysis of corresponding fields. This requirement has emphasized the necessity of a knowledge of computer programming for all the researchers. For research-related purposes, computer programming using Python is one of the best choices. This course is designed with the purpose to make students acquainted with programming using python and make them comfortable to deal with computational data analysis.

Course Objectives
- Help them conceptualize basic theories of computer programming
- Make the students understand fundamental components of python programming
- To develop skills for writing computer programs using all necessary branches of Python
- Accumulate basic ideas about data structures and data manipulations

Course Contents:
Laboratory works based on theory classes and basic problem solving from rosalind.info using Pycharm, Jupyter, and Anaconda IDEs.
Computer Basics: Concept on Computer Hardware, Software and its classification, Compiler vs Interpreter. **Using the Python Interpreter:** Invoking the Interpreter, Argument Passing, Interactive Mode, The Interpreter, and Its Environment, Source Code Encoding; **An Informal Introduction to Python:** Using Python as a Calculator- Numbers, Strings, Lists. First Steps Towards Programming; **More Control Flow Tools:** if Statements, for Statements, The range() Function, break and continue Statements, and else Clauses on Loops, pass Statements, Defining Functions; **More on Defining Functions:** Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations, **Intermezzo:** Coding Style; **Data Structures:** More on Lists- Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, The del statement, Tuples and Sequences, Sets, Dictionaries, Looping Techniques, More on Conditions, Comparing Sequences and Other Types; **Modules:** More on Modules- Executing modules as scripts, The Module Search Path, Compiled" Python files, Standard Modules, The dir() Function, Packages- Importing * From a Package, Intra-package References, Packages in Multiple Directories, matplotlib, numpy, other common necessary packages; **Input and Output:** Fancier Output Formatting, Old string formatting, **Reading and Writing Files**: Methods of File Objects, Saving structured data with JSON; **Errors and Exceptions:** Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions. **Classes:** A Word About Names and Objects, Python Scopes and Namespaces, Scopes and Namespaces Example, A First Look at Classes, Class Definition Syntax, Class Objects, Instance Objects, Method Objects, Class and Instance Variables, Random Remarks, Inheritance, Multiple Inheritance, Private Variables, Odds and Ends, Iterators, Generators, Generator Expressions.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Implement knowledge of Python for writing computer programs
CLO 2	Design solutions of real-life problems using necessary components of Python
CLO 3	Identify errors from a program and use exception handlers to handle errors and exceptions
CLO 4	Implement Object Oriented Programming and modular concepts
CLO 5	Design basic data structures to solve efficient data storage issues
CLO 6	Apply knowledge of programming in data analysis and manipulation

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department
Textbook:
1. Learning Python, By Mark Lutz, 5th Edition
2. Think Python, By Allen B. Downey
3. The Python Tutorial, Official documentation of Python

Course Title: Python Programming Lab	**Credit:** 3 credits
Course Code: CSE 216	**Contact hours:** 6 Hours/Week

Rationale:
In this current world, most of the research works require computational data analysis of corresponding fields. This requirement has emphasized the necessity of knowledge of computer programming for all the researchers. For research-related purposes, computer programming using Python is one of the best choices. This course is designed with the purpose to make students acquainted with programming using python and make them comfortable to deal with computational data analysis.

Objectives:
- Help them conceptualize basic theories of computer programming
- Make the students understand fundamental components of python programming
- To develop skills for writing computer programs using all necessary branches of Python
- Accumulate basic ideas about data structures and data manipulations
- To help them to apply the knowledge of programming for data storage, manipulation, and presentation

Course Contents:
Computer Basics: Concept on Computer Hardware, Software and its classification, Compiler vs Interpreter.
Using the Python Interpreter: Invoking the Interpreter, Argument Passing, Interactive Mode, The Interpreter, and Its Environment, Source Code Encoding; **An Informal Introduction to Python:** Using Python as a Calculator- Numbers, Strings, Lists. First Steps Towards Programming; **More Control Flow Tools:** if Statements, for Statements, The range() Function, break and continue Statements, and else Clauses on Loops, pass Statements, Defining Functions; **More on Defining Functions:** Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations, **Intermezzo:** Coding Style; **Data Structures:** More on Lists- Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, The del statement, Tuples and Sequences, Sets, Dictionaries, Looping Techniques, More on Conditions, . Comparing Sequences and Other Types; **Modules:** More on Modules- Executing modules as scripts, The Module Search Path, Compiled" Python files, Standard Modules, The dir() Function,

Packages- Importing * From a Package, Intra-package References, Packages in Multiple Directories, **Input and Output:** Fancier Output Formatting, Old string formatting, **Reading and Writing Files**: Methods of File Objects, Saving structured data with JSON; **Errors and Exceptions:** Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions. **Classes:** A Word About Names and Objects, Python Scopes and Namespaces, Scopes and Namespaces Example, A First Look at Classes, Class Definition Syntax, Class Objects, Instance Objects, Method Objects, Class and Instance Variables, Random Remarks, Inheritance, Multiple Inheritance, Private Variables, Odds and Ends, Iterators, Generators, Generator Expressions. **Python Numpy:** Numpy intro, creating arrays, array indexing, array slicing, data types, array shape, array iterating, array join, array split, array search, array sort, array filter, random, ufunc. **Python Pandas:** Pandas series, DataFrames, Read CSV, Read JSON, Analyzing Data, Correlations, Plotting. **Python Matplotlib:** Intro, Pyplot, Markers, Line, Subplots, Scatter, Bars, Histograms, Pie Charts.

Course Learning Outcome: After the successful completion of this course students will be able to

CLO1	Implement knowledge of Python for writing computer programs
CLO2	Design solutions of real-life problems using necessary components of Python
CLO3	Identify errors from a program and use exception handlers to handle errors and exceptions
CLO4	Implement Object Oriented Programming and modular concepts
CLO5	Design basic data structures to solve efficient data storage issues
CLO6	Apply knowledge of programming in data analysis and manipulation
CLO7	Apply knowledge of programming for different graphical data representation techniques

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department
Textbook:
1. Learning Python, By Mark Lutz, 5th Edition
2. Think Python, By Allen B. Downey
3. The Python Tutorial, Official documentation of Python

Course Title: Computer Architecture	**Credits:** 3.0

Course Code: CSE 219*	**Contact hours:** 3 hours/week

Rationale:
Students wishing to build up their career in SWE needs to know architecture about computer and this course will help them learning the basics of computer systems and latest hardware architectures.

Objectives:
- To make the students understand the fundamental technologies and performance evaluation of different computer systems;
- To teach what is the instruction set architecture of a system and variations of ISA in different systems;
- To describe how computer performs arithmetic operations;
- To facilitate necessary knowledge about internal architecture of a processor;
- To teach different levels of memory hierarchy and their management in a system.
- To accumulate basic ideas about fundamental technologies on multicore and multiprocessing system and their application.

Course Contents:
Introduction to Computer Architecture: Overview and history; Cost factor; Performance metrics and evaluating computer designs. **Instruction set design:** Von Neumann machine cycle, Memory addressing, Classifying instruction set architectures, RISC versus CISC, Micro programmed vs. hardwired control unit. **Memory System Design:** Cache memory; Basic cache structure and design; Fully associative, direct, and set associative mapping; Analyzing cache effectiveness; Replacement policies; Writing to a cache; Multiple caches; Upgrading a cache; Main Memory; Virtual memory structure, and design; Paging; Replacement strategies. **Pipelining:** General considerations; Comparison of pipelined and non-pipelined computers; Instruction and arithmetic pipelines, Structural, Data and Branch hazards. **Multiprocessors and Multi-core Computers:** SISD, SIMD, and MIMD architectures; Centralized and distributed shared memory- architectures; Multi-core Processor architecture. **Input/output Devices:** Performance measure, Types of I/O device, Buses and interface to CPU, RAID. **Pipelining:** Basic pipelining, Pipeline Hazards. Parallel Processing.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	understand design and architecture of processor, multiprocessor, multi-core processor, distributed systems
CLO 2	understand and develop logic for instruction set architecture
CLO 3	describe the fundamental technologies related to SISD, SIMD, and MIMD architectures
CLO 4	discuss organization, hierarchy and management of different levels of memory and I/O systems
CLO 5	differentiate between Basic Pipelining, Pipelining Hazards and Parallel Processing
CLO 6	evaluate performance of cache memory management, replacement policies and cache writing
CLO 7	analyze cache effectiveness of memory structure.

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Textbook
1. Computer Architecture and Organization by John P.Hayes.
2. Computer Organization and Design: The hardware / software interface by David A.Patterson and John L.Hennessy.

Course Title: Introduction to Computing Application	**Credits:** 3.0
Course Code: CSE 302*	**Contact hours:** 6 hours/week

Rationale
This is a dedicated lab course that aims to make students familiar with the basic computing applications that are frequently used in real life. The main focus of this course is to introduce the familiar applications (i.e. Microsoft Excel, Microsoft Access, SPSS) of Spreadsheet analysis, Database, and Statistical analysis. It covers preparing and working with spreadsheets, formatting spreadsheets, creating charts, using tools, and printing worksheets. It also covers the basics of databases, table designs, indexing, creating queries, creating forms, and printing reports. As part of statistical analysis, it also includes the introduction of statistical analysis, operation commands, data definition and manipulation commands, and procedure commands. All of these will help students to have a good understanding and hands-on experience with the above mentioned basic applications.

Course Objectives
- To assist students in developing a good understanding of the vastly used basic computing applications.
- To help students understand the basic concepts of spreadsheets, databases, and statistical analysis.

- To facilitate the basic knowledge about the workflows of the applications like Microsoft Excel/Access, SPSS.
- To assist students in developing their hands-on skills on the mentioned applications.
- To make students capable of applying their knowledge on the mentioned areas in real life.

Course Contents:
Spreadsheet Analysis: Introduction (Spreadsheet & its Applications, Menus & Toolbars), Working with Spreadsheets (Converting files to different formats, Importing, Exporting, Spreadsheet addressing, Computing data, Mathematical operations, Using formulas), Formatting Spreadsheets (Border & shading, Highlighting values, Visibility, Sorting, Filtering, Validation, Consolidation, Subtotal), Creating Charts (Selecting charts, Formatting charts, label, scaling, etc.), Using Tools (Error Checking, Spell Checks, Macros), Printing worksheet. **Database Applications:** Introduction (Database concepts, Tables, Queries, Forms, Reports), Working with Databases (Creating Tables, Table Design, Indexing, Entering data, importing data), Creating Queries (SQL statements, Setting relationship, using wizards), Creating Forms, Creating & printing reports. **Statistical Analysis:** Introduction, Operation commands, Data definition, manipulation commands, and procedure commands like LIST, DESCRIPTIVES, FREQUENCIES, CROSSTABS, T-TEST, ANOVA, REGRESS, etc.

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	Explain the basic concepts of a spreadsheet.
CLO 2	Prepare a working spreadsheet with all necessary functionalities.
CLO 3	Use spreadsheets to smartly store and manipulate large data.
CLO 4	Explain the concepts of Database (tables, queries, mapping, indexing, forms, etc.)
CLO 5	Design databases.
CLO 6	Create a working database with tables and relations, and make queries in the database.
CLO 7	Describe the basic concepts of statistical analysis.
CLO 8	Analyze (calculate mean/mode/median, regression, correlation, interpolation, etc.) a large amount of statistical data and generate analysis reports using applications like SPSS.

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of the corresponding department

Course Title: Python Programming Lab	**Credit:** 3 credits
Course Code: CSE316	**Contact hours:** 6 Hours/Week

Rationale:
In this current world most of the research works require computational data analysis of corresponding fields. This requirement has emphasized the necessity of knowledge on computer programming for all the researchers. For research related purposes computer programming using Python is one of the best choices. This course is designed with the purpose to make students acquainted with programming using python and make them comfortable to deal with computational data analysis.

Objectives:
- Help them conceptualize basic theories of computer programming
- Make the students understand fundamental components of python programming
- To develop skills for writing computer programs using all necessary branches of Python
- Accumulate basic ideas about data structures and data manipulations
- To help them to apply the knowledge of programming for data storage, manipulation and presentation

Course Contents:
Computer Basics: Concept on Computer Hardware, Software and its classification, Compiler vs Interpreter.
Using the Python Interpreter: Invoking the Interpreter, Argument Passing, Interactive Mode, The Interpreter and Its Environment, Source Code Encoding; **An Informal Introduction to Python:** Using Python as a Calculator- Numbers, Strings, Lists. First Steps Towards Programming; **More Control Flow Tools:** if Statements, for Statements, The range() Function, break and continue Statements, and else Clauses on Loops, pass Statements, Defining Functions; **More on Defining Functions:** Default Argument Values, Keyword Arguments, Arbitrary Argument Lists, Unpacking Argument Lists, Lambda Expressions, Documentation Strings, Function Annotations, **Intermezzo:** Coding Style; **Data Structures:** More on Lists- Using Lists as Stacks, Using Lists as Queues, List Comprehensions, Nested List Comprehensions, The del statement, Tuples and Sequences, Sets, Dictionaries, Looping Techniques, More on Conditions, . Comparing Sequences and Other Types; **Modules:** More on Modules- Executing modules as scripts, The Module Search Path, Compiled" Python files, Standard Modules, The dir() Function, Packages- Importing * From a Package, Intra-package References, Packages in Multiple Directories, **Input and Output:** Fancier Output Formatting, Old string formatting, **Reading and Writing Files**: Methods of File Objects, Saving structured data with json; **Errors and Exceptions:** Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User-defined Exceptions, Defining Clean-up Actions, Predefined Clean-up Actions. **Classes:** A Word About Names and Objects, Python Scopes and Namespaces, Scopes and Namespaces Example, A First Look at

Classes, Class Definition Syntax, Class Objects, Instance Objects, Method Objects, Class and Instance Variables, Random Remarks, Inheritance, Multiple Inheritance, Private Variables, Odds and Ends, Iterators, Generators, Generator Expressions. **Python Numpy:** Numpy intro, creating arrays, array indexing, array slicing, data types, array shape, array iterating, array join, array split, array search, array sort, array filter, random, ufunc. **Python Pandas:** Pandas series, DataFrames, Read CSV, Read JSON, Analyzing Data, Correlations, Plotting. **Python Matplotlib:** Intro, Pyplot, Markers, Line, Subplots, Scatter, Bars, Histograms, Pie Charts.

Course Learning Outcome: After the successful completion of this course students will be able to

CLO1	Implement knowledge of Python for writing computer programs
CLO2	Design solutions of real life problems using necessary components of Python
CLO3	Identify errors from a program and use exception handlers to handle errors and exceptions
CLO4	Implement Object Oriented Programming and modular concepts
CLO5	Design basic data structures to solve efficient data storage issues
CLO6	Apply knowledge of programming in data analysis and manipulation
CLO7	Apply knowledge of programming for different graphical data representation techniques

Mapping of Course Learning Outcomes to Program Learning Outcomes
According to the PLO of corresponding department

Textbook:
1. Learning Python, By Mark Lutz, 5th Edition
2. Think Python, By Allen B. Downey
3. The Python Tutorial, Official documentation of Python

Course Title: Computer Networking	**Credits:** 3.0
Course Code: CSE 317*	**Contact hours:** 3 hours/week

Rationale:
The goal of the course is to teach the fundamental concepts of Networking and Communication Engineering. For those interested in specializing in Communication Engineering, it provides the essentials on which later courses are built.

Objectives:
- To facilitate necessary knowledge about different components and their respective roles in a data communication system
- Acquaint students with the basic tools to analyze the layered architecture of communication protocols.
- To make the students understand the TCP/IP protocol suite and the OSI model.
- Help them conceptualize basic theories in different encoding techniques, multiplexing techniques and switching.
- To explain and implement the concepts and techniques in error detection and correction.

Course Contents:
Introduction: Data communications, Networks, Internet, Protocols and Standards. **Network Models:** OSI Model, TCP/IPProtocol suite, Addressing. **Data and Signals:** Analog and Digital data, Analog and Digital Signals, Time and Frequency Domain, Transmission impairments, Data rate limits, Performance. **Digital Transmission:** Digital-to-Digital Conversion, Analog-to-Digital Conversion, Transmission Modes. **Analog Transmission:** Digital-to-Analog Conversion, Analog-to-Analog Conversion. **Multiplexing and Spread Spectrum:** FDM, WDM, TDM, STDM, Digital Subscriber Line, FHSS, DSSS. **Transmission Media:** Guided and Unguided Media. **Switching:** Circuit switching, Packet switching. **Data Link Layer:** Error Detection and Correction, Data Link Control, Framing, Flow and Error Control. **Multiple Access:** CSMA, CSMA/CD, CSMA/CA, FDMA, TDMA, CDMA.
Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	understand different types of signal related to data communication, their transmission, representation, conversion, limitations and all other relevant information
CLO 2	discuss fundamental concepts of data communication and its components, different types of network, protocol layering and responsibilities of different layers for different protocols
CLO 3	differentiate between Digital-to-Digital Conversion and Analog-to-Digital Conversion
CLO 4	*use Analog-to-Digital and Digital-to-Digital conversion to process Digital Signal Processing*
CLO 5	*analyze resource sharing techniques called multiplexing for transmitting multiple signals through a single channel and identify techniques of different types of multiplexing*
CLO 6	*examine error detection and correction rates, Data Link control and Framing.*

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PLO of corresponding department

Textbook

1. Data Communications and Networking by Behrouz A. Forouzan
2. Data and Computer Communications by W Stallings, Macmillan

Course Title: Computer Networking Lab	**Credits:** 1.5
Course Code: CSE 318*	**Contact hours:** 3 hours/week

Rationale:

The goal of the course is to teach the fundamental concepts of Networking and Communication Engineering. For those interested in specializing in Communication Engineering, it provides the essentials on which later courses are built.

Objectives:

- To facilitate necessary knowledge about the different components and their respective roles in a data communication system
- Acquaint students with the basic tools for analyzing the layered architecture of communication protocols.
- To help the students understand the TCP/IP protocol suite and the OSI model.
- To Accumulate basic ideas about different encoding techniques, multiplexing techniques and switching.
- To Foster the analytical and critical ability of the students to implement the concepts and techniques in error detection and correction.

Course Contents:

Lab experiments and tasks:

1. Implement different Line Coding Schemes using a programming language or MATLAB.
2. Analyze different Analog Transmission Techniques using ANACOM/MODICOM.
3. Implement various Error Detection and Correction techniques using a programming language.
4. Evaluate the performance of the ARQ protocols using a Data Link Layer Protocol Simulator

Course Learning Outcomes: After the successful completion of the course, the student will be able to-

CLO 1	understand the mechanism of LAN and TCP/IP in a local area
CLO 2	differentiate among twisted-pair cable, coaxial cable, and fiber optic cable that are used in communication engineering
CLO 3	implement different error detection and correction techniques, such as Hamming code, CRC, checksum etc.
CLO 4	implement Line Coding Schemes using a programming language or MATLAB
CLO 5	evaluate the performance of the ARQ protocols using a Data Link Layer Protocol Simulator
CLO 6	analyze Analog Transmission Techniques using ANACOM/MODICOM
CLO 7	analyze the mechanism of Peer-to-Peer Network in a network system

Mapping of Course Learning Outcomes to Program Learning Outcomes

According to the PLO of corresponding department

Textbook

1. Data Communications and Networking by Behrouz A. Forouzan
2. Data and Computer Communications by W Stallings, Macmillan

Ordinance for the Graduate Program at SUST

Formation of Graduate Study Committee (GSC) will be the prerequisite to start a Graduate program in any Discipline. The GSC will be headed by the Head of the Discipline/ Institute consisting of all professors/Associate Professors of the discipline concerned with a minimum number of 3 professors/Associate Professors. When Professors and Associate Professors are not available in the discipline, the required number of Professors, Associate Professors will be included from the relevant Discipline/Institute by the proposal of the Board of Advanced Studies (BAS) & the Academic council.

1. Introduction

1.1 The graduate program consists of Masters (General), Masters (Thesis), M.Sc. (Engineering), Masters of Philosophy (M.Phil.) and Ph.D. degrees.
1.2 A graduate program may also be offered by a discipline in some specified field in collaboration with other disciplines.
1.3 Any student with (i) 4 year Bachelors degree (ii) 3-year Bachelor and 1-year Masters Degree or (iii) 5-year Bachelor of Architecture degree from a recognized university is eligible to get admitted into the graduate program at SUST.
1.4 Notification for the admission process will be published every year.
1.5 After admission every student will be assigned to a student advisor/supervisor from among the teachers of his/her discipline to guide him/her throughout the academic program.

2. Qualification

2.1 Masters and M.Phil.
2.1.1 Any student with a Bachelors degree from SUST is eligible for admission to the Masters (General) Program.
2.1.2 Any student with a CGPA of 3.25 or more from SUST is eligible for admission to the Masters (Thesis), M.Phil. (Engineering) or M.Phil. Program.
2.1.3 Four-year Graduates from other recognized universities and institutions with a CGPA of 3.25 or more can apply for admission to the Masters (Thesis), M.Phil. (Engineering) or M.Phil. Program. A candidate who passed under course system and seeks admission to M.Phil program has to have First class in Masters or 50% marks in Masters and at least 2nd division in all public examination.
2.1.4 Any student registered for Masters (General) or Masters (Thesis) may transfer to the M.Phil. program, offered by the relevant discipline, if he/she can maintain a CGPA of 3.25 or more during the first two semesters.
2.1.5 The GSC of a discipline will decide if a student from a related discipline will be allowed to apply to the graduate program of that discipline. In these cases if necessary the GSC may ask the candidate to take extra undergraduate/graduate courses to ensure the basic foundation.

2.2 Ph.D.

2.2.1 Candidates with Masters (Thesis), M.Phil. or M.Sc. (Engineering) Degrees are eligible for application for Ph.D. and will be selected after a written and/or viva voce examination and the proper evaluation of academic records by the GSC. A candidate who passed under course system and seeks admission to Ph.D. program has to have First class in Masters or 50% marks in Masters and at least 2nd division in all public examination.
2.2.2 A Masters (Thesis) an M.Phil. or an M.Sc. (Engineering) student may be transferred to the Ph.D. program after the completion of first two semesters with a CGPA 3.25 and the recommendation of his/her supervisor certifying satisfactory progress of research work and with the approval of the GSC and BAS.
2.2.3 The following candidates are eligible for direct admission to Ph.D. if they have a CGPA of 3.25 or more at Bachelors and Masters Level and 3.00 or equivalent in all public examinations. (i) University teachers with two years teaching experience and one publication in standard academic journals. (ii) Teachers of colleges with three years of teaching experience and one publication in a standard academic journal (iii) Researchers of recognized research organizations with three years of research experience and at least three publications in standard academic journals. (iv) Candidates with an M.Phil degree.

3. Admission

3.1 Masters and M.Phil.
3.1.1 If a SUST graduate has the required qualifications he/she can be admitted to the Masters program (General, Thesis or Engineering) as per the recommendation of the GSC.
3.1.2 The candidates for Masters (Thesis and Engineering) and M.Phil. will be selected for admission after a written and/or viva voce examination conducted by the GSC. Full time teachers of SUST are not required to sit for the admission test. GSC will then recommend the candidates for admission to the academic council through the BAS. During the process of admission each candidate shall be assigned by the appropriate GSC and approved by BAS a supervisor from among the teachers of the relevant discipline/institute not below the rank of an associate professor or an assistant professor with a Ph.D. / M.Phil. / M.S.

3.2 Ph.D.
3.2.1 A candidate for admission to the Ph.D. degree program will apply in the prescribed form to the head of the discipline or the director of institute along with the recommendation from possible supervisor(s). The supervisor must be of the rank of professor or associate professor.
3.2.2 After approval from the GSC, the application will be forwarded to the BAS for the approvals of the supervisor and co-supervisors (if any). Each candidate shall have not more than two co-supervisors; one co-supervisor may be from outside SUST. After careful scrutiny of the research proposal BAS will send it to the Academic Council for final Approval.
3.2.3 If necessary a change of supervisor must also be approved by the BAS and the Academic Council.

4. Registration

4.1 Every selected candidate will be registered with the University and enrolled as a full time or if allowed, part time student with payment of prescribed fees and dues before the commencement of each semester.
4.2 A student has to register for at least 50% or maximum 150% credits of the courses at every semester in the prescribed syllabus. But for attaining degree in the last semester above mentioned restrictions will not be followed.
4.3 A candidate may be admitted or change his status into part time student with prior approval of the university and a written consent from the serving organization. A part time student may be assigned a minimum of 6 credit hours per semester.
4.4 A full time student must register for a minimum of 1(2.0 Credits) hours per semester. A full time student shall not be allowed to be employed as a part time employee in other organizations. However he/she may be employed as teaching/research assistant at the University. A Ph.D. candidate shall have to be a full time student for at least one year during his/her Ph.D. work.
4.5 The registration for the Ph.D. degree will remain valid for a period of four years, and can be renewed for a further period of two years.

5. Academic Regulations

5.1 Duration

5.1.1 The minimum duration for the Masters, M.Sc. (Engineering), M.Phil. and Ph.D. degrees will be as followed:

Degree	Duration of Completion	Required Credits
Masters (General)	2 Semesters	Minimum 24
Masters (Thesis)	3 Semesters	36
M.Phil. / M.Sc. (Engg.)	4 Semesters	48
Ph.D.	6 semesters	72

5.1.2 Minimum duration of M.Phil will be 4 Semesters for students who completed 3 years Bachelors and 1 year Masters degree. Minimum duration of M.Phil will be 2 semesters for students who completed 4 years Bachelors and 1 year Masters degree.

5.2 Credit Requirement

5.2.1 For the graduate program a full time student has to register for at least 1(2.0 Credits) each semester. For course work 1 credit means one hour of contact hour per week and for research or project work 1 credit hour means at least three hours per week. A student will be allowed to take theoretical course and research work simultaneously. Once the course requirement is completed, for the research work a graduate student has to register for "independent study" as credit/no-credit basis to fulfill the 1(2.0 Credits) per semester requirement.

5.3 Course Requirement

5.3.1 Syllabus committee for the graduate program will be comprised of the GSC members and two external members from other universities nominated by the Dean.
5.3.2 Every year the syllabus committee will design the graduate level courses for the respective disciplines and recommend the courses for approval of the Academic Council through the School and BAS. GSC can review the curriculum from time to time and recommend any change to the syllabus committee as may be considered necessary.

Masters and M.Phil.
5.3.3 Every Masters (general, thesis and engineering) and M.Phil. student has to complete at least 16 hours of theory course work during the first two semesters. GSC will propose the required courses to the students with consultation of respective supervisors. The course work for M.Phil Program may be reduced and relaxed according to the recommendation of GSC. In that case the duration may be reduced up to 1 year.

Ph.D.
5.3.4 The GSC may suggest courses, if felt necessary, for the Ph.D. students.

5.4 Research Work Requirement

5.4.1 Research work for thesis shall be carried out under the supervision of the supervisor. Co-supervisors from within or outside the discipline / Institute may be appointed, if necessary. The topic of research proposal shall be approved by the BAS after the completion of the required course credits within six months/one year for M.Phil. / Ph.D. on the recommendation of the Head of the Discipline/Institute. A Ph.D. student must submit a progress report of his work to the supervisor(s) at the end of the every semester who will present it to BAS.
5.4.2 The Ph.D. student will give at least one public seminar talk conducted by GSC at the Discipline / Institute every year on a topic of his own field of research.
5.4.3 The research work must be carried out in this University or at a place approved by the supervisor in consultation with the GSC.

6. Conduct of Examinations

6.1 Course Examination

6.1.1 The examination committee will conduct the course examinations as per the examination ordinance of graduate program.

6.2 Thesis Submission

6.2.1 The title of the thesis has to be approved by the BAS on the recommendation of the Head of the Discipline / Institute. For Masters/M.Phil. it has to be done at least three months and for Ph.D. it has to be done at least six months before submitting.

6.2.2 Every student shall submit to the supervisor required number of type written copies of his thesis in the approved forMATon or before a date to be fixed by the Head of the Discipline/ Institute in the consultation with the supervisor concerned.
6.2.3 The student shall declare that the research work was done by him/her and has not submitted elsewhere for other purpose (except for publication).
6.2.4 The thesis should demonstrate an evidence of satisfactory knowledge in the field of research undertaken by the student.

6.3 Masters Thesis Examination
6.3.1 There is no thesis requirement for Masters (General). The project (if any) and the thesis for Masters (Thesis) and will be evaluated as per the examination ordinance of graduate program.

6.4 M.Phil. / M.Sc. (Engineering) Thesis Examination
Thesis Evaluation
6.4.1 The academic council will, on the basis of the suggestion of the GSC and recommendation of the BAS, appoint for every thesis an examination committee consisting of two examiners of whom at least one shall be from outside this University.
6.4.2 The examiners of thesis will either accept it or reject it for the degree and then individually and separately submit one copy of their reports in sealed covers to the controller of examination and another copy to the GSC Chairman. The majority decision will be considered as the final result.
6.4.3 If a thesis is adjudged inadequate for the award of the degree, the candidate will be allowed to resubmit his thesis within six months. If the candidate fails to resubmit or the thesis is adjudged inadequate again the examiners may recommend Masters (general) degree and the controller of examination will place such recommendation before the BAS for the approval of academic council.

Oral Examination and Open Presentation
6.4.4 The GSC in consultation with the supervisor shall suggest, to the Vice Chancellor through BAS, a committee of three members for oral examination consisting of: (i) Convener: Thesis supervisor (ii) A Professor in relevant field from outside the University (iii) One of the thesis examiners.
6.4.5 If any examiner is unable to accept the appointment or has to relinquish his appointment before/ during the examination, the Vice-Chancellor shall appoint another examiner in his place as per the recommendation of GSC.
6.4.6 After the oral examination the convener will send a consolidated report to the controller of examinations stating clearly whether the award of the degree is recommended, who will in turn place it to BAS for the approval of the Academic Council.
6.4.7 In case a candidate performs unsatisfactorily in oral examination even though the thesis is adjudged adequate the examiners may recommend to the Academic Council that the candidate may be permitted to appear at another oral examination within six months from the first oral examination. No candidate shall be allowed to appear at the oral examination of the same thesis for more than two times.

Recommendation for Degree
6.4.8 After completion for the viva-voce examination, the convener of the viva examination committee will send a consolidated report, stating clearly whether the award of the degree is recommended, to the Controller who will in turn place it to BAS for the approval of the academic council.

6.5 Ph.D. Thesis Examination
Thesis Evaluation
6.5.1 The academic council will, on the basis of the suggestion of the GSC and recommendation of the BAS, appoint for every thesis an examination committee consisting of three examiners of whom one shall be the supervisor and the other two from outside this University and at least one from a university from abroad
6.5.2 One of the three examiners will be appointed by the academic council as the convener of the examination committee.
6.5.3 The examiner of thesis will individually and separately submit one copy of their reports in sealed covers to the controller of examination and another copy to the convener. Every examiner will have to explicitly state whether the award of the Ph.D. degree is recommended or not. The recommendations of all the three examiners must be explicit, unambiguous and unanimous for the award of the degree.
6.5.4 If a thesis is adjudged inadequate for the award of the Ph.D. degree, the candidate will be allowed to resubmit his thesis after six months with proper modification. If the candidate fails to resubmit or the thesis is adjudged inadequate again the examiners may recommend the award of M.Phil. or M. S. degree and the controller of examination will place such recommendation before the BAS for the approval of academic council.

Oral Examination and Open Presentation
6.5.5 On receipt of the unanimous opinions of the examiners, the convener shall fix a date and a venue and suggest, to the Vice Chancellor through BAS, a committee of three members for oral examination consisting of the convener, supervisor/co-supervisor and a thesis examiner. At least one of them has to be from outside the university.
6.5.6 If any examiner is unable to accept the appointment or has to relinquish his appointment before/during the examination, the Vice-Chancellor shall appoint another examiner in his place as per the recommendation of GSC.
6.5.7 In case a candidate is unable to satisfy the viva voice Board even though the thesis is adjudged adequate the Board may recommend to the Academic Council that the candidate may be permitted to appear at another oral examination after a lapse of six months from the first oral examination. No candidate shall be allowed to appear at the oral examination of the same thesis for more than two times.

Recommendation for Degree

6.5.8 After completion of the viva voce examination, the convener will send a consolidated report to the controller of examinations stating clearly whether the award of the degree is recommended, who will in turn place it to BAS for the approval of the Academic Council.

7. Award of the Degree

7.1 Masters

7.1.1 Students will be awarded his/her degree as per the recommendation of GSC chairman after the completion of his required credits.

7.2 M.Phil. and Ph.D.

7.2.1 The vice chancellor shall place the reports of the Oral Examination committee for consideration of the academic council which shall recommend to the Syndicate for the award of the degree.

7.2.2 A hard copy of the thesis accepted by the academic council incorporating any correction and changes suggested by the examination committee shall be preserved in the central library of the university and the corresponding electronic version shall be preserved in the archive.

8. Academic Fee

8.1 To be decided by the Academic Council and the Syndicate.

Ref.:*The clause 4.2 of this Ordinance was approved in the 119th Academic Council.*

Examination Ordinance for the Graduate Program

University authorities will administer and publish the results of Masters, M.Phil. and Ph.D. degree examinations under the graduate program. The graduate program will follow the same academic calendar of the undergraduate program for course delivery, the final examination and publication of results. The graduate courses are comprised of theory and lab courses and where applicable, the thesis for the research works. The evaluation of thesis is conducted as per the Ordinance for the Graduate Program at SUST. The theory and lab courses are conducted by the examination committee.

1. Examination Committee

1.1 The GSC of the Discipline/Institute will form the examination committee as per the rules of the University.
1.2 The examination committee will propose the examination schedule, prepare question papers, help the discipline conducting the examination, prepare results and will resolve the issues that may arise concerning the examination procedure.

2. Examination Dates and Routines

2.1 The examination routines will be designed by the respective disciplines and Head of the disciplines will notify them and send copies to the other relevant disciplines and to the office of the Controller of the Examinations.

3. Theory Courses

3.1 Distribution of Marks
A student will be continuously evaluated during the semester through tests, assignments, mid-semester examinations, viva etc. conducted by the course teachers, and it will contain 30% of total marks. The rest 70% marks will come from the final written examination at the end of that semester.

3.2 Class Performance
After the end of the classes, the course teachers will make three copies of mark-sheets showing the marks from class participation and assignment and mid semester examination. He/she will display one copy in the notice board, send one sealed copy to the chairman of the examination committee and another sealed copy to the controller of examination.

3.3 Question Setting and Moderation
3.3.1 The examination Committee will appoint two question setters for each course at least four weeks before the date of commencement of the examination and inform the Controller of examination. The controller of examination will send the necessary papers to the question setters and the examiners. If a question setter or examiner declines the responsibility, he/she will return all the papers and the examination committee will suggest an alternative question setter or examiner.

3.3.2 The chairman of the examination committee will receive all the manuscript of question papers; if no manuscript is received within the specified time the committee will suggest an alternative question setter.
3.3.3 After receiving all the question papers the examination committee will moderate the question papers. Moderation will not be invalid if any member be absent during moderation. For the disciplines of the school of Applied Sciences and Technology the questions will be divided in two groups in the question paper so that two examiners can evaluate the answer script simultaneously. The examination committee will be responsible for the preparation of the necessary editing and printing of the question papers.

3.4 Final Examination
3.4.1 The controller of examination will be responsible to print the blank answer scripts, mark sheets and other relevant forms and will make necessary arrangements, so that these are available during the conduct of examination in the examination hall in due time .

3.5 Evaluation of Answer Script
3.5.1 The answer scripts from the disciplines of Applied Science and technology will be evaluated by two examiners simultaneously, of whom one should preferably the course teacher. The answer scripts from the disciplines of other school of studies will be evaluated by two examiners separately, of whom one should preferably the course teacher. The examiners will examine the scripts thoroughly, mark the scripts properly and grade legibly within the specified time. The examiners will send a sealed copy of mark-sheet to the controller of examination and one sealed copy to the chairman of the examination committee.
3.5.2 The examination committee will assign members from the committee to scrutinize the answer scripts and if any discrepancy is found the committee will make the necessary arrangements to fix the problem and inform the controller of examination.
3.5.3 If the difference between marks given by two examiners be 20% or more than 20% GSC will recommend a third examiner for approval by the V.C and marks given by 3rd examiner & the marks of the first or 2nd examiner which ever is nearest to this will be considered for the average marks .

4. Lab Courses

4.1 Every lab course will be assigned to at least two course instructors and they will grade the students through continuous evaluation.
4.2 For the projects, Masters (Thesis), Industrial assignments, monographs etc. the supervisor will give an overall assessment which will count as 30% of the total marks. Evaluation of the report by two external examiners, who is not involved in supervision/co-supervision will count as another 30% of the marks. The remaining 40% will come from the presentation and viva voce conducted by the examination committee. During viva-voce examination the supervisor or co-supervisor, if present, will not participate in marking.

5. Publication of Result

5.1 Three original tabulation sheets will be prepared by the tabulators and checked by all the members and signed by the tabulators and members of the examination committee. The tabulation sheets will contain the grade point average obtained in the specific semester. The tabulation sheets will be sent to the Controller of Examinations for his signature and approval by the Vice-Chancellor.
5.2 The Controller of Examination shall keep up to date record of all the grades obtained by the student in individual Academic Record Card. Grades shall be announced by the Controller of Examination at the end of each semester.

Grade and grade points:
5.3 The letter grade and grade point will be awarded as follows:

Numerical Grade	Letter Grade	Grade Points
80% Or above	A+	4.00
75% to less than 80%	A	3.75
70% to less than 80%	A-	3.50
65% to less than 70%	B+	3.25
60% to less than 65%	B	3.00
55% to less than 60%	B-	2.75
50% to less than 55%	C+	2.50
45% to less than 50%	C	2.25
40% to less than 45%	C-	2.00
Less than 40%	F	0.00

6. Security and Ethics

6.1 Everyone involved in the process of examination has to guard the security of the question papers, examination grades and the final results. An examinee can never try to influence the examiners and any such attempt has to be brought to the controller of examination.
6.2 A student may never be asked a question so that he is hurt because of his religious or ethnic background.
6.3 If some one involved in the examination process has the following relatives as examinee he/she should immediately inform in to the authority: (a) Husband/wife, (b) Son/Daughter, (c) Brother/Sister, (d) Brother-in-Law/ Sister-in-Law (e) Son-in-Law/ Daughter-in-Law, (f) Nephew/ Niece, (g) Uncle/ Aunt, (h) First Cousins.

Department of Computer Science and Engineering
Graduate Program
Session 2020-2021

The syllabus for the Graduate Program in CSE covers the requisite courses for the following degrees:

(1) Masters (General)
(2) Masters (Thesis)

(1) Masters (General) and (2) Masters (Thesis) Programs:

Masters (General) in CSE are spreader over two semesters and Master (Thesis) in CSE are spreader over three semesters.The department will fix the courses of two semesters for Masters (General) students including labs and viva-voce considering the minimum requirement of 24 credits for the degree. For the Masters (Thesis) students, the courses of semester-I will be identical of the Masters (General) students. The graduate studies committee (GSC) will approve the courses of semester-II and the theses for semester-III considering the minimum requirement of 36 credits for the Masters (Thesis) degree.

Courses are categorized for three semesters are as follows:

Semester I

Course No	Course Title	Hours/Week Theory and Lab	Credits
CSE5**	Option I	3.00	3.00
CSE5**	Option II	3.00	3.00
CSE5**	Option III	3.00	3.00
CSE500	Project (For Masters (General) Only)	9.00	3.00
CSE601	Independent Study I	9.00	3.00
	Total	**18.00**	**12.00**

Semester II

Course No	Course Title	Hours/Week Theory and Lab	Credits
CSE5**	Option IV	3.00	3.00
CSE5**	Option V	3.00	3.00
CSE5**	Option VI	3.00	3.00
CSE550	Project (For Masters General Only)	9.00	3.00
CSE602	Independent Study II	9.00	3.00
	Total	**18.00**	**12.0**

Semester III

Course No	Course Title	Hours/Week Theory and Lab	Credits
CSE603	Independent Study III	36.00	12.00
	Total	**36.00**	**12.00**

Options

CSE 511 Digital Image Processing
CSE 513Computer Vision
CSE 515Pattern Recognition
CSE 517 Speech Processing
CSE 519 Automatic Speech Recognition

CSE 521Machine Learning and Data Mining
CSE 523 Information Retrieval
CSE 525 Bioinformatics Algorithm Analysis
CSE 527 Sequence Analysis and Genomics

CSE 529 Advanced Computer Networking
CSE 531 Wireless Sensor Networks
CSE 533 Petri Net Theory and System Modeling
CSE 535 Quantum Mechanics and Quantum Computing
CSE 537 Natural Language Processing
CSE 539 Machine Translation

CSE 541 Neural Networks and Fuzzy Systems
CSE 543 Computer System Architecture
CSE 545 Parallel Algorithm
CSE 547 Algorithm and Complexity Analysis
CSE 549 Graph Theory

CSE 551 Distributed Computing System

CSE 553 Advanced Software Engineering
CSE 555 Software Quality Assurance
CSE 557 Advanced Database System with Emerging Application
CSE 559 Computer Simulator and Modeling

CSE 561 Advanced VLSI Design
CSE 563 Robotics and Intelligence System
CSE 565 Research Methodology in Computer Science & Engineering
CSE 567 Contemporary course on Computer Science and Engineering
CSE 569 Big Data Analytics
CSE 571 Introduction to Cryptography and Information Security
CSE 573 Applied Cryptography and Cryptographic Protocol

CSE511 (3 Credits) DIGITAL IMAGE PROCESSING
3 Hours/Week
Digital image fundamentals and transforms: Elements of visual perception, Image sampling and quantization Basic relationship between pixels, Basic geometric transformations-Introduction to Fourier Transform and DFT , Properties of 2D Fourier Transform – FFT – Separable Image Transforms -Walsh – Hadamard – Discrete Cosine Transform,Haar,Slant–Karhunen–Loeve-transforms.
Image enhancement techniques: Spatial Domain methods- Basic grey level transformation, Histogram equalization, Image subtraction, Image averaging, Spatial filtering: Smoothing, sharpening filters, Laplacian filters, Frequency domain filters-Smoothing,Sharpening-filters,Homomorphic-filtering.
Image restoration: Model of Image Degradation/restoration process – Noise models – Inverse filtering -Least mean square filtering – Constrained least mean square filtering – Blind image restoration – Pseudo inverse – Singular value decomposition.
Image compression: Lossless compression: Variable length coding – LZW coding – Bit plane coding- predictive coding-DPCM.Lossy Compression: Transform coding – Wavelet coding – Basics of Image compression standards: JPEG, MPEG,Basics of Vector quantization.
Image segmentation and representation: Edge detection – Thresholding - Region Based segmentation – Boundary representation: chair codes- Polygonal approximation – Boundary segments – boundary descriptors: Simple descriptors-Fourier descriptors – Regional-descriptors–Simple,descriptors-Texture

Textbook:
1. Rafael C Gonzalez, Richard E Woods 2nd Edition, Digital Image Processing - Pearson Education,2003.

CSE513 (3 Credits) COMPUTER VISION
3 Hours/Week
Image Formation Models: Monocular imaging system, Orthographic & Perspective Projection, Camera model and Camera calibration, Binocular imaging systems
Image Processing and Feature Extraction: Image representations (continuous and discrete), Edge detection, Motion Estimation, Regularization theory, Optical computation, Stereo Vision, Motion estimation, Structure from motion
Shape Representation and Segmentation: Deformable curves and surfaces, Snakes and active contours, Level set representations, Fourier and wavelet descriptors, Medial representations, Multiresolution analysis
Object recognition: Hough transforms and other simple object recognition methods, Shape correspondence and shape matching, Principal component analysis, Shape priors for recognition,

Textbook:
Computer Vision - A modern approach, by D. Forsyth and J. Ponce, Prentice Hall
Robot Vision, by B. K. P. Horn, McGraw-Hill.

CSE515 (3 Credits**)** PATTERN RECOGNITION
3 Hours/Week
Pattern Recognition: introduction, importance; Statistical and Neural Pattern Recognition: Bayesian classifier, Bayes decision theory, discriminant functions and decision surfaces; Bayesian classifier for normal distributions; Linear classifiers: discriminant functions and decision hyperplanes, Perceptron algorithm and its variants, Kesslerâ€™s construction; Nonlinear classifiers: two and three layer perceptrons, backpropagation algorithm and its variants; Template matching: optimal path searching techniques, dynamic programming methods, correlation based matching and 2D log search algorithm for image matching; Context dependent classification: Viterbi algorithm, channel equalization, observable and hidden Markov models, three problems of HMM and their application in speech recognition; Syntactic Pattern Recognition: introduction to Syntactic Pattern Recognition, grammar-based approach, parsing, graph-based approach; Unsupervised classification: basic concepts of clustering, proximity measures, categories of clustering algorithms, sequential clustering algorithms.

Text Books:

1. R.O.Duda, P.E.Hart and D.G.Stork, Pattern Classification, John Wiley, 2001
2. S.Theodoridis and K.Koutroumbas, Pattern Recognition, 4th Ed., Academic Press, 2009
3. C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006

CSE517 (3 Credits) SPEECH PROCESSING
3 Hours/Week
Fundamentals of speech processing (familiarity with waveforms, spectra, spectrograms, resonance, formants, human speech production and perception, perceptually-motivated frequency scales, time vs. frequency representations; conversion between the two, the Fourier transform, source-filter model of speech, hands on experience via xwaves), speech recognition (components of a typical recognizer, parameterization of the speech signal, dynamic time warping, distance measures, the Hidden Markov Model, the generative model paradigm, simple probability theory, conditional and joint probabilities, Bayes' theorem, Gaussian probability density function, continuous density HMMs, monophone models with Gaussian observation densities, Viterbi algorithm for recognition, training from fully labeled data, Viterbi training, bigram language models), speech synthesis (components of a typical text-to-speech synthesizer, text analysis, phonology, finite-state automata, POS tagging, lexicon, phrasing, accents, F0, learning from data, CART models, waveform generation, concatenative methods - TD-PSOLA and linear prediction, F0 and duration modification).

CSE519 (3 Credits) AUTOMATIC SPEECH RECOGNITION
3 Hours/Week
Signal analysis for ASR, Statistical pattern recognition (Bayes decision theory, Learning algorithms, Evaluation methods, Gaussian mixture model, and EM algorithm), Hidden Markov Models (HMM), Context-dependent models, Discriminative training, Language models for LVCSR (large vocabulary continuous speech recognition), Decoding, Robust ASR (Robust features Noise reduction, Microphone arrays), Adaptation (Noise adaptation, Speaker adaptation/normalization, Language model adaptation), Speaker recognition, History of speech recognition, Advanced topics (Using prosody for ASR, Audio-visual ASR, Indexing, Bayesian network)

CSE521(3 Credits) MACHINE LEARNING AND DATA MINING
3 Hours/Week
Introduction, Learning to classify: Decision trees, Dealing with numeric attributesOver fitting, Bayesian methods, Naive Bayes learning, Bayesian Networks, Nearest neighbor methods.
Learning to predict numbers: Linear regression, Backpropagation networks, Nearest neighbor methods.
Evaluating learning procedures, Basic Statistics and Experimental Design.
Learning to form clusters:k-means algorithm, Agglomerative hierarchical clustering, Cobweb, Neural nets- competitive learning and Kohonen nets.
Finding association rules: a priori algorithm.
Learning what to do: Reinforcement learningLearning from sequential data, Hidden Markov Models, Multiple models, Evolutionary computation.
Algorithms for the lab: K Nearest Neighbor Classifier, Decision Trees, Model Selection and Empirical Methodologies, Linear Classifiers: Perception and SVM, Naive Bayes Classifier, Basics of Clustering Analysis, K-mean Clustering Algorithm, Hierarchical Clustering Algorithm.

Textbooks:

1. Introduction to Data Mining, P-N Tan, M.Steinbach, V.Kumar, Addison Wesley20062.
2. Data Mining: Concepts and Techniques (2nd ed) Han, J. &Kamber, M., Morgan Kaufmann, 2006
3. Introduction to Machine Learning,Alpaydin, E., MIT Press, 2004
4. Data Mining: Learning Tools and Techniques with Java Implementations (2nd ed), Witten, I. H. &Frank,Morgan Kaufman*n*, 2005

CSE523 (3 Credits) INFORMATION RETRIEVAL
3 Hours/Week
Overview Of The Field
Study Some Basic Concepts Of Information Retrieval. Understanding The Conceptual Model Of An Information Retrieval System. Overview Of Popular Information Retrieval System.

Indexing
Introduce Various Indexing Techniques For Textual Information Items. Inverted Indices, Tokenization, Stemming And Stop Words. Text Processing Methods. Vector Methods. Study Lucene Index Structure. Building An Index.

Retrieval Methods

Study Popular Retrieval Models: Boolean, Vector Space, Binary Independence, Language Modeling. Thesauruses And Cluster Analysis. Other Commonly-Used Techniques Include Relevance Feedback, Pseudo Relevance Feedback, And Query Expansion. Probabililty Ranking Principle.

Evaluation Of Retrieval Performance
Evaluation Of Retrieval Effectiveness. Measurements: Average Precision, Ndcg, Etc. TREC
Relevant Feedback And Personalization
Query Refinement And Relevance Feedback. Study Basic Techniques For Collaborative Filtering And Recommender Systems. Probabilistic Information Retrieval. Probabilistic Latent Semantic Analysis (PLSA). Personalized Web Search Through Click-Through Data.
Web Search Engine
Architecture Of Information Retrieval Systems. Web Crawlers. Search/Retrieve Web Services. Peer-To-Peer Information Retrieval And Mapreduce; Machine Translation; Online (Web) Advertising; And Learning To Rank
Lab Work: Important Algorithm Implementation Relevant To Corse Materials. Building A Search Engine Prototype Using Lucene Indexer.

CSE525(3 Credits) BIOINFORMATICS ALGORITHMS ANALYSIS
3 Hours/Week
Designing algorithm and Programming is an essential skill for bioinformaticians. This course focuses on algorithm and implementation using Perl, the most widely-used programming language for bioinformatics. The implementation part of this course addresses a range of topics, including: how to write short scripts for handling biological data; how to use Bioperl to handle DNA and protein sequence data.
Exhaustive search ,Greedy algorithm, Dynamic Programming, Divide and Conquer, Graph, Clustering and Trees, Randomized Algorithm, Perl: Syntax, Bioperl**.**

CSE527(3 Credits) SEQUENCE ANALYSIS AND GENOMICS
3 Hours/Week
This module covers classical methods of biological sequence analysis and their applications to the problems of modern biology. It also discusses different aspects of molecular evolution: from sequence to structure and function.
Introduction: Genome Organization & Function, Pairwise Sequence Alignment, Multiple Sequence Alignment, Hidden Markov Models and Profiles, Pattern Recognition, Computational Prediction of Functional Elements in Genomic DNA, Phylogenetic Analysis, Machine learning: NN and SVM, Structural Evolution: Folds and Families, Evolution of Protein Function

CSE529 (3 Credits) ADVANCED COMPUTER NETWORKING
3 Hours/Week
Topics include: layered network architectures, applications, transport and routing, IP version 6, mobile IP, multicasting, session initiation protocol, quality of service, network security, network management, and TCP/IP in wireless networks. An emphasis will be placed on the protocols used in the Internet.

CSE531 (3 Credits) WIRELESS SENSOR NETWORKS
3 Hours/Week
Introduction: applications; Localization and tracking: tracking multiple objects; Medium Access Control: S-MAC, IEEE 802.15.4 and ZigBee; Geographic and energy-aware routing; Attribute-Based Routing: directed diffusion, rumor routing, geographic hash tables; Infrastructure establishment: topology control, clustering, time synchronization; Sensor tasking and control: task-driven sensing, information-based sensor tasking, joint routing and information aggregation; Sensor network databases: challenges, querying the physical environment, in-network aggregation, data indices and range queries, distributed hierarchical aggregation; Sensor network platforms and tools: sensor node hardware, sensor network programming challenges; Other state-of-the-art related topics.

CSE533(3 Credits) PETRI NET THEORY AND SYSTEMMODELING
3 Hours/Week
Definition and types of Petri nets, Terms and notations marking, Importance of net theory, Transition firings, Practical modeling examples, Siphons and traps, Live ness and safeness, Behavioral properties, Deadlocks and siphons, Structural properties, Stochastic Petri Net (SPN).

CSE535(3 Credits) QUANTUM MECHANICS AND QUANTUM COMPUTING
3 Hours/Week
The two slit experiments; Measurements and observable; Commutation of observations; linear operators; Eigenvalue equations. Complementary Principle; Physical postulates of Quantum mechanics; Wave function and its interpretation; probability density and probability current density, Eigenstates; Orthonormality of eigenstates; Principle of superposition; Probability amplitudes and overlap integrals; Wave packets and uncertainty principle, The Schrodinger wave equation and one dimensional potential problems -- particle in a potential box, potential step, tunneling through potential barrier, rectangular potential well; Linear harmonic oscillator; Angular momentum: Orbital angular momentum; Rotation operator, Spherical harmonics, Solution of the Schrodinger equation for Hydrogen atom.
Quantum mechanics: State vectors in Hilbert space; bra and ket notations; operators and their representation; transformation theory; Schrodinger, Heisenberg, and Dirac representations. Pauli's exclusion principle and spin matrices. Theory of scattering : Two-body systems; scattering by spherically symmetric potentials; partial- wave analysis; Stationary perturbation theory; time dependent perturbation theory; Identical particle : Symmetric and antisymmetric wave functions;
Application of Quantum Mechanics in Quantum Computation

CSE537 (3 Credits) NATURAL LANGUAGE PROCESSING
3 Hours/Week
Introduction;Word Modeling: Automata and Linguistics, Statistical Approaches and Part of Speech Tagging;Linguistics and Grammars; Parsing Algorithms; Parsing

Algorithms and the Lexicon;Semantic;Feature Parsing; Tree Banks and Probabilistic Parsing**;** Machine Translation; Evolutionary Models of Language Learning and Origins.
Processing of words, Phrase structure parsing, Semantic Interpretation with Phrase Structure Grammars related Lab Work.

CSE 539 (3 Credits) MACHINE TRANSLATION
3 Hours/Week
Introduction to MT: What is MT? Why MT matters?; Application; Challenges: Linguistic divergences between English and Bangla; Approaches: The Voquoius triangle; History;
Evaluation: Human: Adequacy, Fluency; Automatic: BLEU, TER, NIST
Approaches: Rule-based MT (RBMT): Transfer-based, Interlingua; Example-based MT (EBMT); Statistical MT (SMT): Phrase-based SMT (PBMT), Factored SMT, Hierarchical SMT, Syntax-based SMT; Neural MT (NMT);
PBMT: Translation model: Word-based models, IBM models, Expectation Maximization (EM) training, Asymmetric alignments, Phrase-based models, Alignment symmetrization, Extracting bilingual phrases from a symmetric word alignment, Estimating phrase translation probabilities, Bidirectional translation probabilities, Bidirectional lexical weighting, Word penalty, Phrase penalty, Lexicalized reordering model; Language model: n-gram LM, Neural probabilistic LM (NPLM); Log-linear models; Tuning and Decoding: Beam-search stack decoder;
NMT: Basic Encoder-Decoder, Attentional mechanism, Self-attentional/Transformer mechanism, Deep architecture;
Processing: Error Analysis; Pre-processing/Pre-editing; Post-processing/Post-editing; Re-ranking; Adding linguistics: Lexical, Morphological, Syntactical; Techniques for Low-resource scenario;
Resources: Corpora: Parallel (SUPara, OPUS, etc.), Monolingual (SUMono, Europarl, etc.), Comparable; Toolkits: Moses, OpenNMT, Tensor2Tensor, Nematus; Systems: Google, shu-anubad; NLP tools: banglaSketch;

CSE541 (3 Credits) NEURAL NETWORK AND FUZZY SYSTEMS
3 Hours/Week
Fundamentals of Neural Networks; Back propagation and related training algorithms; Hebbian learning; Cohonen-Grossberg learning; The BAM and the Hopfield Memory; Simulated Annealing; Different types of Neural Networks: Counter propagation, Probabilistic, Radial Basis Function, Generalized Regression, etc; Adaptive Resonance Theory; Dynamic Systems and neural Control; The Boltzmann Machine; Self-organizing Maps; Spatiotemporal Pattern Classification, The Neurocognition; Practical Aspects of Neural Networks.
Basic Concepts of Fuzzy set theory; Fuzzy numbers; Aggregation operations of Fuzzy sets; Fuzzy Rules and Fuzzy Reasoning, Fuzzy Inference Systems, Regression and Optimization, Supervised Learning Neural Networks, Neuro-Fuzzy Modeling, ANFIS, Neuro-Fuzzy Control, ANFIS Applications.

CSE543(3 Credits) COMPUTER SYSTEM ARCHITECTURE
3 Hours/Week
This course examines the structure of modern computer systems. We explore hardware and technology trends that have led to current machine organizations, then consider specific features and their impact on software and performance. These may include superscalar issue, caches, pipelines, branch prediction, and parallelism. Midterm and final exams, team project, homework, in-class exercises.

CSE545 (3 Credits) PARALLEL ALGORITHMS
3 Hours/Week
Introduction to design and analysis of parallel algorithms, computational models and complexity classes. Parallel algorithms for various problems including: basic arithmetic, sorting, searching, selection, graph theory, matrix computations, combinatorial enumeration, optimization, computational geometry, and numerical analysis will also be extensively studied in the course.

CSE 547 (3 Credits) ALGORITHM AND COMPLEXITY ANALYSIS
3 Hours/Week
Basic Concepts, Problem Definition, encoding, instance, size, decision & optimization problems. Turing Machines - Deterministic and Nondeterministic polynomial Reducibility. Classification of Problems - P, NP, NP-complete, NP-hard. Examples of similarly defined easy and difficult problems. Cook's Theorem, several important NP-complete problems, techniques of proving a problem, NP-complete restriction, component design, local replacement problems of intermediate complexity, problems beyond NP.

CSE549(3 Credits) GRAPH THEORY
3 Hours/Week
Introduction, Fundamental concepts, Trees, Spanning trees in graphs, Distance in graphs, Eulerian graphs, Digraphs, Matching and factors, Cuts and connectivity, k-connected graphs, Network flow problems, Graph coloring: vertex coloring and edge coloring, Line graphs, Hamiltonian cycles, Planar graphs, Perfect graphs.

CSE551 (3 Credits) DISTRIBUTED COMPUTING SYSTEMS
3 Hours/Week
Distributed object systems, Retrieving and caching of distributed information, Distributed data replication and sharing, Performance issues, Algorithms for deadlock detection, Concurrency control and synchronization in distributed system, Models for distributed computation, Networking facilities and resource control and management methods in network and distributed operating systems, Collaborative applications, Wide area network computing, Web based commerce, Agent systems and Market based computing.

CSE553 (3 Credits) ADVANCED SOFTWARE ENGINEERING
3 Hours/Week
Advanced specification and design in UML, component-based software engineering, rapid development processes and techniques, advanced validation and verification methods, configuration management, and other advanced topics.

CSE555 (3 Credits) SOFTWARE QUALITY ASSURANCE
3 Hours/Week
Definition and concept of software quality assurance (SQA); quality models; specification of quality requirements; product development & delivery issues; software development processes & maturity; software quality management process: total quality management, improvement cycle, SQA planning & management, organizing the SQA effort; software verification & validation; typical software development errors; Fagan inspections; software audit; software testing: testing objectives & testing fundamentals, testing theory, coverage criteria, equivalence class testing, value-based testing, decision table, syntax & state transition testing, statement & path testing, branch & condition testing, data flow testing, thread-based testing, integration & integration testing, system testing; testing in object-oriented systems; test tools & test automation; test management; problem reporting & corrective action.

CSE557 (3 Credits) ADVANCED DATABASE SYSTEMS WITH EMERGING APPLICATIONS
3 Hours/Week
Object Oriented Database; Data Model, Design, Languages; Object Relational Database: Complex data types, Querying with complex data types, Design; Distributed Database: Levels of distribution transparency, Translation of global queries to fragment queries, Optimization of access strategies, Management of distributed transactions, Concurrency control, Reliability, Administration; Parallel Database: Different types of parallelism, Design of parallel database; Multimedia Database SystemsBasic concepts, Design, Optimization of access strategies, Management of Multimedia Database Systems, Reliability; Database Wire-housing/Data mining: Basic Concepts and algorithms.

CSE559 (3 Credits) COMPUTERSIMULATION AND MODELING
3 Hours/Week
Introduction to simulation, Examples of Simulations, Statistical modelsProbability, random variables and their properties, mathematical expectation, specific discrete and continuous random variates (Poisson, exponential, etc.). Simulation tools, random number and variate generation, event serialisation and time advance algorithms; process and resource classes, Performance measures, model instrumentation and result presentation. Simple stochastic processes - discrete time Markov chains, continuous time Markov processes; Poisson process, Birth and Death process and their application to the simple (e.g. M/M/1) queues. More advanced queuing theory - multi-server queues, non-Markovian queues, networks of queues. mean value analysis (analytic derivation of throughput, utilisation, mean queue size and delay). Applications - case studies in computer systems and networks using analysis and simulation, advanced simulation software.

CSE561(3 Credits) ADVANCED VLSI DESIGN
3 Hours/Week
Review of CMOS logic circuits; impact of fabrication issues on design; high speed switching circuits; high performance memory structures; advanced clocking strategies and clock distribution; performance optimization; deep sub-micron design issues; ASIC design flow: logic synthesis, placement and routing; design verification; low power design. Students will learn and participate in the process of design, simulation and layout of a complex digital system.

CSE563(3 Credits) ROBOTICS AND INTELLIGENT SYSTEMS
3 Hours/Week
Overview and Preliminaries, Articulated Robots; Coordinates and Transformations, Mobile Robots, Personal Assistants, and Games;Rigid-Body Dynamics, Dynamic Systems, Sensors and Actuators, Dynamic Effects ofFeedback Control, Analog and DigitalControl Systems, Introduction to Optimization, Optimal Control, Least-Squares Estimation and Numerical Optimization, Monte Carlo Evaluation andEvolutionary Algorithms, Formal Logic andComputing; Predicate Calculus;1st-order Logic, and Fuzzy Sets; Probability and Statistics, Multivariate Statistics andStochastic Control; Stochastic, Robust, and Adaptive Control, Classification of Data Sets, Introduction to Neural Networks, Training Neural Networks, Machine Learning andKnowledge Representation, Task Planning andMulti-Agent Systems.

CSE565 (3 Credits) RESEARCH METHODOLOGY FOR COMPUTER SCIENCE AND ENGINEERING
3 Hours/Week
IntroductionPaper Writing and Publication, Theory of Science, Paper Search, Research Ethics, Research Methods in Computer Science and Engineering (introduction), Research Methods, Computer Science History, Paper Ppublishing and Rreviewing, Measured-based research methods in Computer Engineering.

CSE567 (3 Credits) CONTEMPORARY COURSE ON COMPUTER SCIENCE AND ENGINEERING
3 Hours/Week
This course covers a contemporary title in Computer Science and Engineering decided by the department.

CSE569 (3 Credits) BIG DATA ANALYTICS
3 Hours/Week
Introduction to Big Data: Defining Big Data, Delivering business benefit from Big Data; **Storing Big Data:** Analyzing your data characteristics; **Overview of Big Data stores:** Data models, Hadoop Distributed File System, HBase, Hive, Cassandra, Hypertable, Amazon S3, BigTable, DynamoDB, MongoDB, Redis, Riak, Neo4J; **Selecting Big Data Stores:** Choosing the correct data stores based on your data characteristics, Moving code to data, Implementing polyglot data store solutions, Aligning business goals to the appropriate data store; **Processing Big Data:** Integrating disparate data stores, Employing Hadoop MapReduce, The building blocks of Hadoop, MapReduce, Handling streaming data; **Tools and Techniques to Analyze Big Data:** Abstracting Hadoop MapReduce jobs with Pig, Performing ad hoc Big Data querying with Hive, Creating business value from extracted data; **Developing a Big Data Strategy:** Defining a Big Data strategy for your organization, Enabling analytic innovation, Implementing a Big Data Solution.

CSE571(3 credits) INTRODUCTION TO CRYPTOGRAPHY AND INFORMATION SECURITY
3 Hours/Week
The objective of this course is to introduce graduate students to the mathematical concepts of cryptography and its applicability to information security. Basic programming skills are critical prerequisite.
Classical Cryptography. Number Theory. Shannon's Theory, Perfect Secrecy and the One-Time Pad. Block Ciphers and Stream Ciphers. Hash Functions and Message Authentication. The RSA Cryptosystem and Factoring Integers. Public-Key Cryptography and Discrete Logarithms. Identification Schemes and Entity Authentication. Key Distribution. Key Agreement Schemes. Secret Sharing Scheme, Zero-knowledge Proof. Secure Multi-party Computation.

Textbook
1. Introduction to Cryptography with coding theory–LawrenceC. Washington and Wade Trappe
2. Cryptography and Network Security –WilliamStallings

CSE573(3 credits) APPLIED CRYPTOGRAPHY AND CRYPTOGRAPHIC PROTOCOL
3 Hours/Week
In this course students will use various cryptographic primitives to implement security protocols. The course consists of lectures and extensive programming assignments. Cryptography concepts from CSE571 will be essential. Basic programming skills are critical prerequisite.
Abstract Syntax Notation One (ASN.1). Security Protocols. Kerberos, Key Exchange & Management Protocol, Public Key Infrastructure (PKI), Transport Layer Security (TLS), Certificate Revocation List (CRL), Online Certificate Status Protocol (OCSP), Smart Cards, Tor, Trusted Platform,Bitcoin& Blockchain Systems.

Textbook
1. Handbook of Applied Cryptography – AlfredJ. Menezes, Paul C. van Oorschot and Scott A. Vanstone
2. Applied Cryptography: Protocols, Algorithms, and Source Code in C– BruceSchneier

Project and Independent Study

CSE500/550 (3 Credits) PROJECT
3 Hours/Week
Project work based on theory courses.

CSE601 (3 Credits) INDEPENDENT STUDY I
3 Hours/Week
Research work approved by GSC and BAS

CSE602 (3 Credits) INDEPENDENT STUDY II
3 Hours/Week
Research work approved by GSC and BAS

CSE603 (3 Credits) INDEPENDENT STUDY III
3 Hours/Week
Research work approved by GSC and BAS