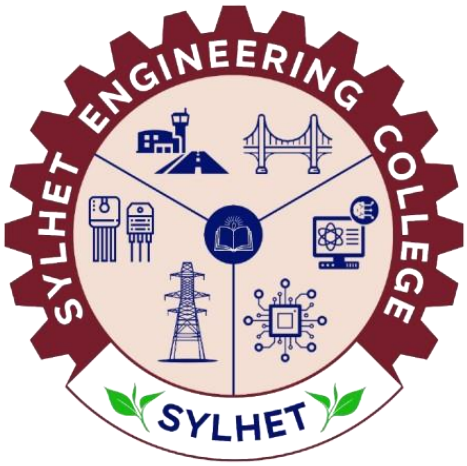
CURRICULUM
UNDERGRADUATE 2025-2026



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SYLHET ENGINEERING COLLEGE
SYLHET, BANGLADESH.

Editorial Committee

Md. Abu Naser Mojumder
Head
Department of Computer Science & Engineering
Sylhet Engineering College

Dr. Mohammad Shahidur Rahman
Professor
Department of Computer Science & Engineering
Shahjalal University of Science and Technology

Dr. Md. Mohammad Reza Selim
Professor
Department of Computer Science & Engineering
Shahjalal University of Science and Technology

Khaleda Ferdousi
Associate Professor
Department of Computer Science & Engineering
Sylhet Engineering College

Md Lysuzzaman
Assistant Professor
Department of Computer Science & Engineering
Sylhet Engineering College

Published by

Department of Computer Science & Engineering
Sylhet Engineering College, Sylhet
Bangladesh

CHAPTER 1
GENERAL INFORMATION

1.1 History

Sylhet Engineering College, abbreviated as SEC, the total development process of a Country cannot proceed without the development of science and Technology. Today's world is coming forward to meet the Challenge of the twenty-first Century through the rapid development of science and technology. To meet the Challenge of the 21st Century, the Country needs to produce a huge number of engineers and technologies.

There are five Universities of Engineering and Technology and some public and private Universities in the Country for providing engineering education at the Degree level, which is not sufficient.

In order to meet the present and future job market demand in the country and abroad, it is essential to establish new engineering colleges. It is also essential to expand Engineering/Technological education to ensure proper use of technology.

In the Sylhet Engineering College, each year 180 students will graduate in three degree-level Courses, Such as (i) B.Sc in Civil Engineering, (ii) B.Sc in Electrical and Electronics Engineering, and (iii) B.Sc in Computer Science & Engineering. After obtaining a graduation in Engineering, the graduates will be able to get a job in different industries and Business organizations in the Country and abroad, and also they will be able to make themselves Capable for Self employment.

Sylhet Engineering College is established as a degree-level Engineering College in the public Sector. The examination of the courses will be controlled by the Shahjalal University of Science and Engineering, and the Students of Sylhet Engineering College will be awarded a degree from Shahjalal University of Science and Engineering

1.2 Location

The Sylhet Engineering College campus is located at Tilagarh, Sylhet Sadar. It is a nicely located place. There is a female student's dormitory for 80 female students, and two male student hostels for 320 students. It is close to Sylhet Agriculture University. The College Campus accommodates three Academic Buildings, one Library Building, one Principal's Quarters, etc.

1.3 Undergraduate Studies

Undergraduate Courses are Computer Science & Engineering. Civil Engineering, Electrical and Electronic Engineering,

1.4 Postgraduate Studies and Research

At present Post Graduate studies and research are not included but in future college may be upgraded and will offer Master Degree. The expertise of the College teachers and the laboratory facilities of the Engineering College are also utilized to solve problems and to provide up-to-date engineering and technological knowledge to the various organizations of the country. In future government should make effort to improve its research facilities, staff position and coursed and curricula to meet the growing technological challenges confronting the country.

1.5 Faculties, Departments and Teachers

At present, Engineering College has four teaching departments including three Engineering departments offer B.Sc in Engineering Degree. A total of 40 full time and part time teachers are teaching in these Departments.

CHAPTER 2
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2.1 Historical Background

The Department of Computer Science and Engineering, the first department, is going to start from the academic year 2007-08. At the very beginning, the department will offer only B.Sc Engineering Degree in Computer Science & Engineering. At first, 60 students will be admitted each year to pursue the B.Sc. Engineering Degree.

2.2 Location

The academic building of the Computer Science & Engineering is 4-stored. There are classrooms, different laboratories and a room for the teaching staff. There are also provisions for the departmental library.

2.3 Study Program

The Department of Computer Science and Engineering offers the degree of B.Sc. Engineering. The courses and syllabus followed by this department for the above degrees are the most modern ones like those of advanced countries as well as appropriate to the local needs. The syllabus is so designed as to contain all the necessary study materials so that a graduate can face the engineering problems readily after graduation. The teachers of the department will meet periodically to review the courses and their contents: necessary changes are made to update the needs and trends from time to time.

2.4 Research Activities

CSE SEC provides the highest quality of research at the international level from Bangladesh. Faculty and Students of CSE SEC have strong research involvement. Government and private sectors prefer faculties of CSE SEC for the solutions to their technical and innovative operations.

2.5 Laboratory Facilities

At present there are five different laboratories in the department premises. A brief description of each of the laboratory facility follows.

2.5.1 Software Engineering Laboratory

This laboratory has a total number of 64 workstations and 2 servers with multimedia support. A multimedia projector belongs to this laboratory to facilitate presentations. All the workstations provide Windows and Linux platforms and have important software installed.

2.5.2 Microprocessor & Interfacing Laboratory

The digital laboratory is equipped with modem equipment, a trainer to demonstrate, design and implement various microprocessor-based circuits. This laboratory provides widespread opportunity to gain knowledge about assembler software, complier software and PLC trainer and other devices. This laboratory has a vast number of ICs in stock, starting from simple 74 series chips up to different types of microprocessors and their peripheral chips. There are various Microprocessor Trainer kits such as 8088 based MTS 88.C µkit and 8086 based µkit.

2.5.3 Networking Laboratory

The students will acquire knowledge of network management, establishment and maintenance by using the various networking devices present in this laboratory. The workstations in this laboratory have been loaded with deferent networking software that allows the students to monitor and experiment with different aspects of computer networking.

2.5.4 Communication Laboratory

The communication laboratory is equipped with modem server with network multimedia player and other communication equipment like digital communication trainer, telecommunication trainer, microwave trainer, etc to demonstrate different theories. The communication laboratory provides widespread opportunity to gain knowledge about communication engineering. There is also two servers with 30 work station. All the workstations provide Windows XP and Linux platforms and have important software installed.

2.5.5 Image Processing & Artificial Intelligence Laboratory

This laboratory has 64 high performance workstations with multimedia support. The laboratory has two Flatbed Scanner, two color Digital Video Camera, All the stations are connected with the department LAN. They communicate with an 802.11g/2.4 GHz wireless Access point which is connected to the backbone LAN.

2.6 Library Facilities

There is a provision for small departmental library in the department. The library will be enriched day by day.

2.7 Co-curricular Activities

Students of this department will arrange different co-curricular activities like programming contests, software and hardware project competitions, software fair etc.

2.8 Training

CSE SEC offers professional trainings to students as well as industry personnel for their skill development.

2.9 Consultation Services

For consultation and research the expertise, its teachers and the laboratory facilities are available to other organizations of the country. SEC is not only contributing as the focal point for the development and dissemination of engineering and technological services within the country, but also it is involved to solve complicated practical problems of national importance faced by the planners, engineers and technologists of the country. Highly qualified and diversely experienced consultants of various fields of engineering have been involved in this endeavor. Wide ranges of quality control testing facilities are also available for materials used in various development activities. Expert consultancy services ranging from the analysis, design, evaluation, construction, rehabilitation, etc. are routinely carried out.

CHAPTER 3
RULES AND RFGULATIONS FOR UNDERGRADUATE PROGRAM

3.1. Student Admission

3.1.1 Undergraduate Admission:
The admission committee of the university will conduct the admission process for the 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 subject to the verification of academic records as per the university's rules.

3.1.2 Student Status and Student Level:
Every student has to maintain his/her student status by getting admission paying the necessary fees and registering for required credits every semester. Unless a student graduates early by taking courses in advance, every student has to get admission in every semester, successively. For bookkeeping purpose, a student's level will be expressed by his/her year and semester. A student will be transferred to the 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 graduate.

3.1.3. 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.

3.1.4 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.

3.2. Academic Calendar

3.2.1 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.

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

Classes and Preparatory weeks	16 weeks
Final Examination	04 weeks
Total	20 weeks

These 20 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.

3.3 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.

3.3.1 Course Development:

3.3.1.1 Major and Non-Major Courses:
The 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 in consideration of the needs of that discipline.

3.3.1.2 Syllabus:
Major and Non-Major Courses: The 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-requisites for any course if deemed necessary.

3.3.1.3 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 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.

3.3.2 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

3.3.2.1 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.	CSE	Computer Science and Engineering
2.	EEE	Electrical and Electronic Engineering
3.	MAT	Mathematics
4.	PHY	Physics
5.	IPE	Industrial and Production Engineering
6.	CHE	Chemistry
7.	ENG	English
8.	SS	Social Science
9.	STA	Statistics
10.	BUS	Business Studies

3.3.2.2 Course Number:
The eight-digit number will be used as follows:
(a) First Four Digit: The first four digits of the eight-digit number will correspond to the rules by the INTERNATIONAL STANDARD CLASSIFICATION OF EDUCATION.
(b) Second Two Digit: The second two digits of the eight-digit number will correspond to the year intended for the course recipient.
(c) Third Two Digit: The third two digits of the eight-digit number will be used to identify a course within a particular discipline. These two digits can be used sequentially to indicate follow-up courses. If possible, even numbers will be used to identify laboratory courses.

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

3.3.2.4 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.

3.3.3 Assignment of Credits:

3.3.3.1 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-minute recesses between theory classes). A theory course will have only an integer number of credits.

3.3.3.2 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.

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

Will be assigned by the respective discipline.

3.4. Course Registration

3.4.1 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 performance. 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.

3.4.2 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.

3.4.3 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 course 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.

3.4.4 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.

3.4.5 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.

3.5. Graduation Criteria

3.5.1 Major Degree:

3.5.1.1 Total Credits:

A student has a requirement of 160 credits for graduation.

3.5.1.2 Total Years:

A regular student is expected to graduate in 8 semesters (4 years). A student may graduate in shorter period if s/he is willing to take extra courses systematically. 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.

3.5.1.3 Minimum Credit for a Clearing Graduate:

For a clearing graduate (8th and subsequent semesters) condition for maximum and minimum credit requirements is relaxed.

3.5.1.4 Break in study:

Those students who have not been able to achieve their degrees by participating in the ascertained 12th 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 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., the 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.

3.6. Examination System

A student will be evaluated continuously in the course 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 work, from his/her written reports and grades of examinations designed by the respective course teacher and the examination committee.

3.6.1 Distribution of Marks:

The marks of a given course will be as follows:

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

3.6.1.1 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%.

3.6.1.2 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.

3.6.1.3 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 will have two answer scripts to answer separate questions during the final examination. Two separate examiners will grade the two scripts separately and the marks will be added together to get the final mark.

3.7. Grading System

3.7.1 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

3.7.2 Calculation of Grades

3.7.2.1 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.

3.7.2.2 CGPA:
The 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 rounded value of the third digit after the decimal is nonzero, the second digit will be incremented by one.

3.7.2.3 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.

3.8. Distinction

3.8.1 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.

CHAPTER 4
COURSE REQUIREMENTS FOR UNDERGRADUATE COMPUTER SCIENCE AND ENGINEERING STUDENTS

Vision Statement
The Department of Computer Science and Engineering, SEC, intends to provide an excellent educational environment 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. To create highly skilled computer engineers, capable of doing research and also developing 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 represents 13 weeks of classes, excluding final exams. Semester-wise breakdown of the curriculum structure for 2024-25 session are shown.

Undergraduate students of the Department of Computer Science and Engineering have to follow a particular course schedule which is given in this chapter according to semester-wise distribution of the courses:

SEMESTER-I

Course Number	**Course Title**	**Hours/Week**		**Credit**	**Pre-requisite**
		Theory	Practical/Sessional		
CSE 0613 1133	Structured Programming Language	2.00	---	2.00	
CSE 0613 1134	Structured Programming Language Sessional	--	6.00	3.00	
CSE 0541 1143	Discrete Mathematics	3.00	---	3.00	
EEE 0713 1109	Introduction to Electrical Engineering	3.00	---	3.00	
EEE 0713 1110	Introduction to Electrical Engineering Sessional	---	3.00	1.50	
MAT 0541 1101	Co-ordinate Geometry and Linear Algebra	3.00	---	3.00	
PHY 0533 1103	Physics	3.00	---	3.00	
PHY 0533 1104	Physics Sessional	---	3.00	1.50	
	Total	**14.00**	**12.00**	**20.00**	

SEMESTER-II

Course Number	Course Title	Hours/Week		Credit	Pre-requisite
		Theory	Practical/Sessional		
CSE 0613 1237	Data Structures	3.00		3.00	
CSE 0613 1238	Data Structures Sessional		3.00	1.50	
IPE 0632 1206	Engineering Drawing Sessional	---	3.00	1.50	
CSE 0613 1247	Theory of Computation	2.00	----	2.00	
CHE 0531 1201	Chemistry	3.00	---	3.00	
CHE 0531 1202	Chemistry Sessional		2.00	1.00	
MAT 0541 1203	Calculus	3.00	---	3.00	
ENG 0231 1201	English Language	2.00	---	2.00	
ENG 0231 1202	Communication in English (Practice)	---	2.00	1.00	
SS 0311 1205	Managerial Economics	3.00	---	3.00	
CSE 0610 1250	Project Work I		3.00	1.50	
	Total	**16.00**	**13.00**	**22.50**	

SEMESTER-III

Course Number	Course Title	Hours/Week		Credit	Pre-requisite
		Theory	Practical/Sessional		
CSE 0613 2133	Object Oriented Programming Language	3.00	---	3.00	
CSE 0613 2134	Object Oriented Programming Language Sessional	---	3.00	1.50	
CSE 0541 2157	Numerical Methods	2.00	---	2.00	
CSE 0541 2158	Numerical Methods Sessional with Python	---	2.00	1.00	
CSE 0688 2147	Engineering Ethics and Cyber Law	2.00	---	2.00	
CSE 0613 2134	Introduction to Competitive programming	---	3.00	1.50	
EEE 0714 2111	Electronic Devices and Circuits	3.00	---	3.00	
EEE 0714 2112	Electronic Devices and Circuits Sessional	---	3.00	1.50	
STA 0542 2102	Statistics for Engineers	3.00	---	3.00	
IPE 0632 2105	Management for Engineers	2.00	---	2.00	
	Total	**15.00**	**11.00**	**20.50**	

SEMESTER-IV

Course Number	Course Title	Hours/Week		Credit	Pre-requisite
		Theory	Practical/Sessional		
CSE 0613 2237	Algorithms	3.00	---	3.00	
CSE 0613 2238	Algorithms Sessional	---	3.00	1.50	
CSE 0613 2201	Digital Logic Design	3.00	---	3.00	
CSE 0613 2202	Digital Logic Design Sessional	---	3.00	1.50	
CSE 0714 2279	Computer Architecture	3.00	---	3.00	
BUS 0411 2205	Accounting Fundamentals for IT Professionals	3.00	---	3.00	
MAT 0541 2204	Complex Variables, Laplace's Transforms and Fourier Series	3.00	---	3.00	
CSE 0610 2250	Project Work II		3.00	1.50	
CSE 0610 2290	Viva Voce		2.00	1.00	
	Total	**15.00**	**11.00**	**20.50**	

SEMESTER-V

Course Number	Course Title	Hours/Week		Credit	Pre-requisite
		Theory	Practical/Sessional		
CSE 0612 3163	Database Management System	3.00	---	3.00	
CSE 0612 3164	Database Management System Sessional	---	3.00	1.50	
CSE 0619 3193	Artificial Intelligence	3.00		3.00	
CSE 0619 3194	Artificial Intelligence Sessional		3.00	1.50	
CSE 0612 3159	Web Technologies	2.00	---	2.00	
CSE 0612 3160	Web Technologies Sessional	---	3.00	1.50	
CSE 0714 3177	Microprocessors and Microcontrollers	3.00	---	3.00	
CSE 0714 3178	Microprocessors and Microcontrollers Sessional	---	3.00	1.50	
CSE 0714 3171	Data Communication	3.00		3.00	
CSE 0714 3172	Data Communication Sessional	---	3.00	1.50	
	Total	**14.00**	**15.00**	**21.50**	

SEMESTER-VI

Course Number	Course Title	Hours/Week		Credit	Pre-requisite
		Theory	Practical/Sessional		
CSE 0619 3293	Machine Learning	3.00	---	3.00	
CSE 0619 3294	Machine Learning Sessional	---	3.00	1.50	
CSE 0613 3235	Operating System	3.00	---	3.00	
CSE 0613 3236	Operating System Sessional	---	3.00	1.50	
CSE 0714 3275	Digital Signal Processing	3.00	---	3.00	
CSE 0714 3276	Digital Signal Processing Sessional	---	3.00	1.50	
CSE 0613 3241	Software Engineering and Design Patterns	3.00	---	3.00	
CSE 0613 3242	Software Engineering and Design Patterns Sessional	---	3.00	1.50	
CSE 0611 3230	Technical Writing and Presentation	---	3.00	1.50	
CSE 0610 3250	Project Work III	---	4.00	2.00	
CSE 0610 3290	Viva Voce		2.00	1.00	
	Total	**12.00**	**21.00**	**22.50**	

SEMESTER-VII

Course Number	Course Title	Hours/Week		Credit	Pre-requisite
		Theory	Practical/Sessional		
CSE 0610 4122	Thesis I	---	4.00	2.00	
Or					
CSE 0610 4144	Project I	---	4.00	2.00	
CSE 0612 4161	Cloud Computing	3.00	---	3.00	
CSE 0612 4162	Cloud Computing Sessional	---	3.00	1.50	
CSE 0613 4131	Computer Graphics	3.00	---	3.00	
CSE 0613 4132	Computer Graphics Sessional	---	3.00	1.50	
CSE 0612 4165	Computer Networks	3.00		3.00	
CSE 0612 4166	Computer Networks Sessional		3.00	1.50	
Optional					
CSE 0613 4167	Simulation and Modeling	3.00	---	3.00	
CSE 0613 4168	Simulation and Modeling Sessional		3.00	1.50	
Or					
CSE 0619 4193	Deep Learning	3.00	---	3.00	
CSE 0619 4194	Deep Learning Sessional		3.00	1.50	
Or					
CSE 0612 4165	Compiler Design	3.00	---	3.00	
CSE 0612 4166	Compiler Design Sessional		3.00	1.50	
Or					
CSE 0613 4171	Introduction to Quantum Computing	3.00	---	3.00	
CSE 0613 4172	Introduction to Quantum Computing Sessional		3.00	1.50	
Or					
CSE 0714 4173	Internet of Things	3.00	---	3.00	
CSE 0714 4174	Internet of Things Sessional		3.00	1.50	
	Total	**12.00**	**16.00**	**20.00**	

SEMESTER-VIII

Course Number	Course Title	Hours/Week		Credit	Pre-requisite
		Theory	Practical/Sessional		
CSE 06104266	Thesis II	---	8.00	4.00	
Or					
CSE 06104288	Project II	---	8.00	4.00	
CSE 0613 4247	Cryptography and Network Security	3.00	---	3.00	
CSE 0613 4248	Cryptography and Network Security Sessional	---	3.00	1.50	
CSE 0610 4290	Viva Voce	---	2.00	1.00	
Optional					
CSE 0613 4231	Digital Image Processing	3.00	---	3.00	
CSE 0613 4232	Digital Image Processing Sessional	---	3.00	1.50	
Or					
CSE 0714 4271	Wireless & Mobile Communication	3.00	---	3.00	
CSE 0714 4272	Wireless & Mobile Communication Sessional	---	3.00	1.50	
Or					
CSE 0613 4237	Advanced Algorithm Engineering	3.00	---	3.00	
CSE 0613 4238	Advanced Algorithm Engineering Sessional	---	3.00	1.50	
Or					
CSE 0688 4297	Bio-informatics	3.00	---	3.00	
CSE0688 4298	Bio-informatics Sessional	---	3.00	1.50	
Or					
CSE 0714 4277	VLSI Design	3.00	---	3.00	
CSE 0714 4278	VLSI Design Sessional	---	3.00	1.50	
Or					
CSE 0613 4243	Natural Language Processing	3.00	---	3.00	
CSE 0613 4244	Natural Language Processing Sessional	---	3.00	1.50	
Or					
CSE 0612 4267	Distributed and Parallel Computing	3.00	---	3.00	
CSE 0612 4268	Distributed and Parallel Computing Sessional	---	3.00	1.50	
Or					
CSE 0612 4269	Generative AI and Large Language Models	3.00	---	3.00	
CSE 0612 4270	Generative AI and Large Language Models Sessional	---	3.00	1.50	
	Total	**6.00**	**16.00**	**14.00**	

Summary

Semester	Hours/Week		Credit	Pre-requisite
	Theory	Sessional		
Semester-1	14.00	12.00	20.00	
Semester-2	16.00	13.00	22.50	
Semester-3	15.00	11.00	20.50	
Semester-4	15.00	11.00	20.50	
Semester-5	14.00	15.00	21.50	
Semester-6	12.00	21.00	22.50	
Semester-7	12.00	16.00	20.00	
Semester-8	6.00	16.00	14.00	
Total	**104.00**	**115.00**	**161.50**	

Departmental subjects	:	121.00	credits
Science and other subjects	:	40.50	credits
Total	:	161.50	credits

CHAPTER 5
DETAIL OUTLINE OF UNDERGRADUATE COURSES OFFERED BY THE DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SEMESTER-I

CSE 0613 1133: Structured Programming Language
2 hours in a week, 2.00 Credit
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. **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-

CO 1	Design, implement, and debug C programs using variables, control structures, and bitwise operations to solve computational problems, demonstrating proper use of flowcharts for algorithmic design
CO 2	Construct modular C programs employing functions, recursion, pointers, and structures, while evaluating memory efficiency through dynamic allocation and storage classes.
CO 3	Develop and analyze stack/queue ADTs using arrays and linked lists, implementing searching/sorting algorithms to optimize performance in applications like expression evaluation and file processing.
CO 4	Evaluate and optimize advanced algorithms (e.g., quicksort, recursive backtracking) through dynamic memory management and pointer arithmetic, creating solutions for complex problems like the Tower of Hanoi and system-level programming.

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								
CLO2			X						
CLO3			X						
CLO4		X							

Textbook

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

CSE 0613 1134: Structured Programming Language Sessional
6 hours in a week, 3.00 Credit
Laboratory works based on CSE **0613 1133**
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-

CO 1	Design, implement, and debug C programs using variables, control structures, and bitwise operations to solve computational problems, demonstrating proper use of flowcharts for algorithmic design
CO 2	Construct modular C programs employing functions, recursion, pointers, and structures, while evaluating memory efficiency through dynamic allocation and storage classes.
CO 3	Develop and analyze stack/queue ADTs using arrays and linked lists, implementing searching/sorting algorithms to optimize performance in applications like expression evaluation and file processing.
CO 4	Evaluate and optimize advanced algorithms (e.g., quicksort, recursive backtracking) through dynamic memory management and pointer arithmetic, creating solutions for complex problems like the Tower of Hanoi and system-level programming.

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								
CLO2			X						
CLO3			X						
CLO4		X							

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

CSE 0541 1143 Discrete Mathematics
3 hours in a week, 3.00 Credit

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-

CO 1	Explain the basic concepts of set, algorithm, combinatorics, logic, graphs, trees, and boolean algebra
CO 2	Apply mathematical knowledge and logical reasoning to solve different problems
CO 3	Compare multiple algorithms to solve any specific problems in terms of time complexity
CO 4	Design new problems and expected solutions based on the knowledge they have gathered discussing in a group

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								
CLO2		X							
CLO3		X		X					
CLO4									X

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

EEE 0713 1109: Introduction to Electrical Engineering
3 hours in a week, 3.00 Credit
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	Understand fundamental terms related to electrical circuits.
CLO 2	Summarize the basics of energy storage elements, AC circuit concepts and responses.
CLO 3	Interpret the basic circuit laws and circuit analysis techniques.
CLO 4	Apply circuit analysis techniques to analyze DC and AC circuits.
CLO 5	Design and analyze DC as well as single-phase AC circuits.

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

EEE 0713 1110: Introduction to Electrical Engineering Sessional
3 hours in a week, 1.50 Credit

In this course students will perform experiments to verify practically the theories and concepts learned in **EEE 0713 1109**. 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 the operating principle of various electrical instruments and measuring devices.
CLO 2	Design experiments to interpret different types of circuit analysis theorem and laws.
CLO 3	Explain the idea about complex circuit networks.
CLO 4	Interpret transient responses of the capacitor and inductor circuits.
CLO 5	Demonstrate team-based communication skills and apply these in practical life

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PLO10
CLO1	X			X						
CLO2				X						
CLO3		X								
CLO4				X						
CLO5										X

Recommended Books

1. Fundamental of Electric Circuits – Charles K. Alexander and Matthew N.O. Sadiku

2.Introductory Circuit Analysis by Robert L.Boylestad

MAT 0541 1101 Co-ordinate Geometry & Linear Algebra

3 hours in a week, 3.00 Credit

In an increasingly complex world, mathematical thinking, understanding, and skill are more important than ever. **MAT 0541 1101 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:

Co-ordinate Geometry : Transformation of co-ordinates axes and its uses: Equation of conics and its reduction to standard forms; Pair of straight lines; Homogeneous equations of second degree: Angle between a pair of straight lines; Pair of lines joining the origin to the point of intersection of two given curves, circles; System of circles; Orthogonal circles; Radical axis, radical center, properties of radical axes; Coaxial circles and limiting points; Equations of parabola; ellipse and hyperbola in Cartesian and polar co-ordinates; Tangents and normal , pair of tangents; Chord of contact; Chord in terms of its middle points; Pole and polar parametric co-ordinates; Diameters; conjugate diameters and their properties; Director circles and asymptotes.

Linear Algebra: Introduction to systems of linear equations ;homogeneous systems; Gaussian elimination; 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 ; Real vector spaces and subspaces; linear dependence and independence; spanning set and basis; coordinates and dimension, Change of basis, Rank and Nullity; Inner product spaces; Eigen values and eigen vectors; diagonalization of matrices; Cayley-Hamilton theorem; Euclidean n-space; Linear transformations: Linear transformations; composition of transformations Linear transformations from IRn to IRm; Properties of linear

transformations from IRn to IRm ; Cauchy-Schwartz inequality; orthogonality; orthonormal basis and Gram-Schmidt process.

Course Learning Outcomes (CLO):

After completing the course the students will be able to

CLO 1	Learn how to transform an equation from one axis to another axis .
CLO 2	Understand properties of circle and explain the geometric condition that determine the relation between two or more circle .
CLO 3	Learn properties of Pair of straight line ,Parabola ,Hyperbola and Ellipse.
CLO 4	Solve system of linear equation and determine the size, transpose, inverse, rank, and LU-factorization of a matrix
CLO 5	Understand element and operation of Real vector space and determine span ,basis ,dimension of those element.
CLO 6	Explain the mathematical foundation of eigen value and eigen vector and Understand matrix diagonalization .

Mapping of Course Learning Outcomes to Program Learning Outcomes

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				

Books Recommended:

1. Thomas Finney: Calculus and Analytic Geometry.
2. Howard Anton and Chris Rorres: Elementary linear algebra applications
3. Steven J. Leon: Linear algebra with applications, Prentice Hall, 1998
4. Rahman and Vattacharjee :A Text book on Coordinate Geometry with Vector Analysis

PHY 0533 1103 Physics

3 hours in a week, 3.00 Credit

Rationale:

In this course, Students will be able to gather knowledge of thermal properties of materials and apply The knowledge in different thermal situations. This course will also provide basic knowledge in structure Of matter, wave and oscillations. Physical optics will covered by this course through which students will be familiar with interference, Bi-prism and diffraction.

Objectives

- To learn about thermometer and its construction.
- To learn basic principles of thermodynamics.
- To know wave behavior and Lissajous figure.
- To learn physical optics and problem solving technique.

Course Content:

Heat and Thermodynamics: Principle of temperature measurements: platinum resistance thermometer, thermo-electric thermometer, pyrometer; Kinetic theory of gases: Maxwell's distribution of molecular Speeds, mean free path, equipartition of energy, Brownian motion, Van der Waal's Equation of state, Review of the First Law of thermodynamics and its application, reversible and irreversible processes, Second Law of thermodynamics, Carnot cycle; Efficiency of heat engines, Carnot's Theorem, entropy of reversible and irreversible process.

Structure of Matter: Crystalline and non-crystalline solids, single crystal and polycrystalline solids, unit Cell, crystal systems, co-ordinations number, crystal planes and directions, NaCl and CsCl Structure, packing factor, Miller indices, relation between inter-planar spacing and Miller indices, Bragg's Law.

Waves and Oscillations: Differential equation of a simple harmonic oscillator, total energy and average energy, combination of simple harmonic oscillations, Lissajous figures, spring-mass system, calculation of time period of torsional pendulum, damped oscillation, determination of damping co-efficient, forced oscillation.

Physical Optics : Theories of light; Interference of light, Young's law, double slit experiment; Displacements of fringes and its uses; Fresnel Bi-prism, interference at wedge shaped films, Newton's rings, interferometers, Diffraction of light: Fresnel and Fraunhoffer diffraction, diffraction by single slit.

Course Learning outcomes

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

CLO1: Explain thermometer, kinetic theory of gases, mean free path, Brownian motion, van der walls equation and related problems.

CLO2: Learn basic law of thermodynamics and solve related problems.

CLO3: Understand structure of matter.
CLO4: Learn wave behaviour and calculate wave properties for different situations.
CLO5: Know physical optics and related problems.

Mapping of the CLOs with PLOs

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
CLO4	X	X					X		X
CLO5	X	X	X				X	X	X

Recommended Books

1. Physics for Engineers. Dr.Giasuddin Ahmed
2. Halliday, D. and Resnick, R: physics (Vol.I and Vol II)

PHY 0533 1104 Physics Sessional
3 hours in a week, 1.50 Credit

Laboratory works based on PHY 101.

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 students to carry out some fundamental experiments for finding out the numerical values of some physical parameters based on various laws, principles and theorems of physics.

Course contents:

1. Determination of the value of 'g' gravity by using compound pendulum.
2. Determination of the spring constant and effective mass of a spiral spring.
3. Determination of the focal length of a convex lens.
4. Determination of the mechanical equivalent of heat by electrical method.
5. Determination of the velocity of sound by water tube and tuning fork.
6. Calculation of the Planck's constant using LED.
7. Determination of angle of rotation of a sugar solution using half-shade Polarimeter.
8. Determination of the radius of curvature of a plano-convex lens by Newtons ring method.
9. Determination of specific heat of a liquid by the method of cooling.
10. Comparison of e.m.f of two cells by potentiometer.
11. Determination of Frequency of tuning fork by Melde's apparatus.
12. Determination of refractive index of a prism.

Course learning outcomes

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

CLO 1	investigate the laws and principles of the moment of inertia, simple harmonic oscillation, conservation of energy, etc.
CLO 2	measure the material properties like Young's modulus and modulus of rigidity.
CLO 3	apply the knowledge of electric circuits, underlying physics of the resistance, galvanometer, Wheatstone's bridge, etc. in designing experiments to measure observable quantities.
CLO 4	collect and analyze experimental data, and draw logical conclusions from the data.
CLO 5	calculate the uncertainty of measurements.
CLO 6	work in a group, report scientific findings, and develop communication skills.

Mapping of the CLOs with PLOs

CLO/PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9	PL10
CLO1	X									
CLO2	X									
CLO3	X									
CLO4	X	X	X	X						
CLO5	X			X						
CLO6									X	X

Recommended Books

1. Practical Physics: Dr.Giasuddinahmed & Md.Shahabuddin

SEMESTER-II

CSE 0613 1237 Data Structures
3 hours in a week, 3.00 Credit

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 deqeues
- 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	Define and explain the fundamental data structures such as lists, queues, trees, etc.
CLO 2	Illustrate the concept of algorithm complexity analysis
CLO 3	Compare tradeoffs in the design and implementations of the data structures
CLO 4	Select appropriate algorithms to use in specific applications and apply including real life problem

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

CSE 0613 1238 Data Structures Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on 0613 1237.
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 138.

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

CLO 1	Understand various data representation techniques in the real world.
CLO 2	Implement linear and non-linear data structures.
CLO 3	Analyze various algorithms based on their time and space complexity
CLO 4	Design and develop real-world applications using suitable data structure.

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

IPE 0632 1206: Engineering Drawing Sessional
3 hours in a week, 1.50 Credit

Rationale:

Engineering drawing is the language of the engineers and technicians. Therefore, it is the intent of this course to equip students with the fundamentals of this unique language and to give them the skills necessary to prepare complete, concise, and accurate communications through engineering drawings using Inventor Professional. 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 computer engineering drawing
- students can also develop an understanding of 2D and 3D computer aided drafting with the requirements of good engineering drawings and be able to apply them to their work.
- 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:

Engineering Drwaing:Introduction, Introduction to Graphics Drawing with Microsoft Visio.,Communication, Instruments and their uses, Sketching, Engineering Geometry, Multiviews and Visualization, Auxiliary Views, Pictorial Projections, Dimensioning and Tolerancing Practices, Working Drawings and Assemblies ,First angle and third angle projections, Orthographic drawing, Sectional views. Isometric views, Missing lines and views. **Data Visualization**: Introduction to data visualization, Visualization Tools,Basic Plotting(Line plot - Bar plot - Pie Chart - Scatter Plot - Histogram - Stacked Bar Charts - Sub Plots - Matplotlib, Searborn, Plotly - Seaborn Styles), Applied Visualizations(Box plot - Density Plot - Area Chart - Heat map - Tree map - Graph Networks),

Principles for data visualization(Visual Perception and Cognition - Gestalt's Principles - Tufte's Principles - Applications of Principles of Information Visualization - Dashboard Design), Interactive Visualizations and Animations(Dynamic charts - Dynamic maps - Animation types - 2D, 3D, Motion Animation - Animation Principles - Altair Package - Statistical Visualizations), Introduction to Tableau

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
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.)

Mapping of CLOs with PLOs

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

Books Recommended:

1. Bertoline & Wiebe, Fundamentals of Graphics Communication, , 6th Edition, McGraw-Hill International Edition
2. William McKinney ,Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython
3. Garrett Grolemund & Hadley Wickham, R for Data Science
4. K.V. Reddy, Textbook of Engineering Drawing, BS Publications, India.
5. Thomas E. French, Charles J. Vierck, Robert J. Foster ,Engineering Drawing and Graphic Technology-International Edition, , McGraw-Hill, Inc.1993 ISBN 0-07-022347-5
6. M.B. Shah and B. C. Rana, Engineering Drawing, Dorling Kindersley (India) Pvt Ltd.

CSE 0613 1247 Theory of Computation

2 hours in a week, 2.00 Credit

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

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

CHE 0531 1201: Chemistry
3 hours in a week, 3.00 Credit
Rationale

In order to create global leaders in computer sectors the graduates will be knowledgeable about basic chemistry to perceive engineering materials and their applications. The student with the proficiency of basic phenomenon of chemistry, they will understand and explain scientifically the various chemistry related problems in the engineering field. As this is a support course taken usually in the first year, it will be helpful for the improvement of skill levels pertaining to the use of technology and communication in device fabrication.

Objectives:

- To compare the chemical behavior and physical properties of common substances.
- To solve quantitative problems (stoichiometric) involving chemical formulas and equations.
- To impart knowledge of green chemical technology and its applications.
- To analyze chemical processes involved in engineering arena.
- To evaluate quality of engineering products.
- To enhance the thinking capabilities in line with the modern trends in engineering and technology for pursuing further research.

Course Content:

Atomic structure, Quantum numbers, Electronic configuration, Periodic table; Properties and uses of noble gases; Different types of chemical bonds and their properties; Molecular structure of compounds; Selective organic reactions; Different types of solutions and their compositions; Phase rule, Phase diagram of monocomponent system; Properties of dilute solutions; Thermochemistry, Chemical kinetics, Chemical equilibria; Ionization of water and pH concept; Electrical properties of solution.

Course Learning Outcomes:

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

CLO 1	Illustrate modern concepts of atomic structure and their limitations, correlation of atomic models, demonstration of orbit and orbitals, electron distribution and energy levels, hydrogen spectral lines etc.
CLO 2	Explain the development of the periodic table, comparison of periodic trends in physical and chemical properties of elements in the periodic table.
CLO 3	Calculate the percent composition of a compound and derive empirical formulas from experimental data with the concept and use of different concentration unit, limiting reactant and percent of yield.
CLO 4	Describe different types of chemical bonds and their properties, draw molecular structures of various compounds.
CLO 5	Differentiate organic compounds (alkane, alkene, alkyne etc.) and discuss their properties and reactions.

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**	**PLO8**	**PLO9**
CLO 1	X	X	X						
CLO 2	X	X							
CLO 3	X	X							

CLO 4	X	X	X						
CLO 5	X	X			X				

Books Recommended:

1. General Chemistry; Author: Darrell. D. Ebbing.
2. Physical Chemistry; Author: Peter Atkins.
3. Introduction to Modern Inorganic Chemistry; Author: S.Z. Haider.
4. Organic Chemistry; Author: Morrison and Boyd.

CHE 0531 1202: Chemistry Sessional

2 hours in a week, 1.00 Credit

Rationale

In order to create global leaders in computer sectors the graduates will be knowledgeable about basic chemistry to perceive engineering materials and their applications. The student with the proficiency of basic phenomenon of chemistry, they will understand and explain scientifically the various chemistry related problems in the engineering field. As this is a support course taken usually in the first year, it will be helpful for the improvement of skill levels pertaining to the use of technology and communication in device fabrication.

Objectives:

- To compare the chemical behavior and physical properties of common substances.
- To solve quantitative problems (stoichiometric) involving chemical formulas and equations.
- To impart knowledge of green chemical technology and its applications.
- To analyze chemical processes involved in engineering arena.
- To evaluate quality of engineering products.
- To enhance the thinking capabilities in line with the modern trends in engineering and technology for pursuing further research.

Course Content:

Atomic structure, Quantum numbers, Electronic configuration, Periodic table; Properties and uses of noble gases; Different types of chemical bonds and their properties; Molecular structure of compounds; Selective organic reactions; Different types of solutions and their compositions; Phase rule, Phase diagram of monocomponent system; Properties of dilute solutions; Thermochemistry, Chemical kinetics, Chemical equilibria; Ionization of water and pH concept; Electrical properties of solution.

Course Learning Outcomes:

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

CLO 1	Illustrate modern concepts of atomic structure and their limitations, correlation of atomic models, demonstration of orbit and orbitals, electron distribution and energy levels, hydrogen spectral lines etc.
CLO 2	Explain the development of the periodic table, comparison of periodic trends in physical and chemical properties of elements in the periodic table.
CLO 3	Calculate the percent composition of a compound and derive empirical formulas from experimental data with the concept and use of different concentration unit, limiting reactant and percent of yield.
CLO 4	Describe different types of chemical bonds and their properties, draw molecular structures of various compounds.
CLO 5	Differentiate organic compounds (alkane, alkene, alkyne etc.) and discuss their properties and reactions.

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	PLO8	PLO9
CLO 1	X	X	X						
CLO 2	X	X							
CLO 3	X	X							
CLO 4	X	X	X						
CLO 5	X	X			X				

Books Recommended:

1. General Chemistry; Author: Darrell. D. Ebbing.
2. Physical Chemistry; Author: Peter Atkins.
3. Introduction to Modern Inorganic Chemistry; Author: S.Z. Haider.
4. Organic Chemistry; Author: Morrison and Boyd

MAT 0541 1203 Calculus

3 hours a week, 3.00 Credit

Rationale: Mathematics is the language of science which develops thinking and critical problem solving skills. Differential calculus deals with the calculation of instantaneous rate of change and integral calculus deals with finding out the limit of a summation of the infinitely many small factors. The calculus has wide applications in science, engineering, economics, finance, statistics etc. The content of the course comprises functions, limits,

continuity, derivatives, tangent and normal, different theorems such as Rolle's, Mean value, Taylor's, Leibnitz's and Euler's theorem etc, indefinite and definite integrals and their applications in real life situations. The course also contain the solution of different types of ordinary differential equations which helps to solve complex engineering problems.

Course Objectives:

- Know the basic concept of function and its applications to real – life problems.
- Explore the concepts, properties, and aspects of the differential and integral calculus of single variable functions.
- Learn the concepts of limits, continuity and derivative.
- Learn to finding out the derivative of different type of functions applying the formulae of derivatives.
- Know the application of derivatives to solve maximum and minimum value problems.
- Study various types of integrations for different cases.
- Apply the techniques of integration to solve the real-life oriented problems such as length, areas and volumes etc.
- Learn the solution of different types of ordinary differential equations and their applications.

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; functions of two or three variables; Partial and total derivatives. Maximum and minimum values of functions ; Points of inflexion; Curvature, radius of curvature, center of curvature; Asymptotes, curve tracing.

Integral Calculus: Physical meaning of integration of a function; integration as an inverse process of differentiation; different techniques of integrations; definite integration 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; Higher 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-.

CO 1	Demonstrate knowledge of basic pre-calculus concepts and skills
CO 2	Evaluate limits
CO 3	Recognize continuity and use the properties of continuous functions
CO 4	Find derivatives of algebraic and trigonometric functions using the definition or basic rules of differentiation
CO 5	Develop and practice methods of differential calculus with applications
CO 6	Develop and practice methods of the integral calculus

Mapping of CLO with PLO:

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

Recommended Books

1. Das and Mukherjee: Differential Calculus
2. Das and Mukherjee: Integral Calculus
3. Abu Yousuf: Differential Equations.
4. Abu Yousuf :Differential and Integral Calculus

ENG 0231 1201 English Language

2 hours in a week, 2.00 Credit

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
CLO1									X
CLO2									X
CLO3									X
CLO4									X
CLO5									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

ENG 0231 1202 Communication in English (Practice)
2 hours in a week, 1.00 Credit

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

Grammar: Tense, article, preposition, subject-verb agreement, clause, conditional and sentence structure.

Vocabulary building: Correct and precise diction, affixes, level of appropriateness. Colloquial and standard. informal and formal.

(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

SS 0311 1205 Managerial Economics
3 hours in a week, 3.00 Credit

This 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

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				

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

CSE 0610 1250: Project Work I
3 hours in a week, 1.50 Credit
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	PLO1	PLO2	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			

SEMESTER -III

CSE 0613 2133: Object Oriented Programming
3 hours in a week, 3.00 Credit

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

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/PLO	PLO1	PLO2	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. Intro**d**uction 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.
4. Programming with Java by E Balagurusamy

CSE 0613 2134 Object Oriented Programming Sessional
3 hours in a week. 1.50 Credit
Laboratory works based on CSE **0613 2134**.

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	Write and execute a basic java program using proper syntax, compilation, and execution procedure of Java.
CLO 2	Model classes from real-world problems in terms of objects rather than procedures.
CLO 3	Apply object-oriented programming principles to implement small and medium-scale Java programs with simple graphical user interfaces.
CLO 4	Design Java programs for complex problems, making good use of the features of the language such as classes, inheritance, polymorphism, abstraction, package, and interface.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X								

CLO2		X							
CLO3			X						
CLO4				X					

Textbook

1. Intro**d**uction 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.
4. Programming with Java by E Balagurusamy

CSE 0541 2157 Numerical Methods
2 hours in a week, 2 Credit
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	PLO1	PLO2	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. Numerical Methods for Engineers by Steven C. Chapra, Raymond P. Canale.
2. Introduction to Numerical Analysis by F.B. Hildebrand.

CSE 0541 2158 Numerical Methods Sessional with Python
2 hours in a week, 1.00 Credit

Laboratory works based on CSE 0541 2157.

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	Implement algorithms from a known/given pseudocode for solving numerical analysis problem
CLO 2	Design and/or implement algorithms directly from a given problem statement (without needing any pseudocode)
CLO 3	Implement bug-free and efficient codes against all algorithms
CLO 4	Perform teamwork to solve complex real-world problems and communicate their findings on a written report and/or by oral presentations

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

CSE 0688 2147 Engineering Ethics and Cyber Law
2 hours in a week, 2 Credit

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	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1							X	X	
CLO2							X	X	
CLO3							X	X	
CLO4							X	X	

CSE: 0613 2134 Introduction to Competitive programming
3 hours in a week, 1.50 Credit

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	PLO1	PLO2	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

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.

EEE 0714 2111 Electronic Devices and Circuits
3 hours in a week, 3.00 Credit

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

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							

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

EEE 0714 2112 Electronic Devices and Circuits Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on EEE **0714 2111**.

Rationale
In this course students will perform experiments to verify practically the theories and concepts learned in **EEE 0714 2112**. 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.

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			X		
CLO3	X	X					X	X	
CLO4	X	X	X						
CLO5	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

STA 0542 2102 Statistics for Engineers
3 hours in a week, 3.00 Credit

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

Content 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

IPE 0632 2105 Management for Engineers
2 hours in a week, 2 Credit

Rationale
This course aims to provide engineering students with the basic knowledge of management principles and practices required in engineering and industrial organizations. The course integrates organizational management, personnel management, inventory management, project planning and management, marketing management, and technology management to develop managerial and decision-making skills alongside technical knowledge.

Course Objectives
The objectives of this course are to:
- provide knowledge about the principles and functions of management and organizational structures relevant to engineering organizations;
- develop understanding of personal management, leadership, motivation, teamwork, and professional development;
- familiarize students with inventory management concepts and techniques used in industrial and engineering operations;
- introduce project planning, scheduling, monitoring, and control techniques for engineering projects;
- develop understanding of marketing and technology management concepts for innovation, competitiveness, and sustainable organizational growth.

Course Content
Management and Organization: Definition and evolution of management, Functions of management, Organizational structure and design, Formal and informal organizations, Span of management, Delegation of authority, Job design; **Personnel Management:** Scope and importance, Need hierarchy, Motivation theories; Productivity and satisfaction, Leadership styles, Teamwork and group dynamics, Mitigating conflicts; **Personnel Development:** Recruitment and selection, Training and development, Performance appraisal, Wages and incentive, Job evaluation and job enrichment; **Inventory Management:** Inventory concepts, Inventory costs, Inventory control techniques, ABC analysis, EOQ model; **Project Planning and Management:** Project concepts, Project life cycle, Project scheduling, CPM, PERT, Budgeting, Monitoring and control; **Marketing Management:** Concept, Pricing strategies, Sales promotion; Patent laws; **Technology Management:** Technology and innovation management, Technology life cycle, Technology transfer.

Course Learning Outcomes (COs)
After successful completion of the course, students will be able to:
CO1: explain the principles, functions, and organizational structures used in engineering and industrial organizations.
CO2: apply concepts of personnel management, motivation, leadership and teamwork in professional environments.
CO3: evaluate and apply inventory management techniques for effective inventory control and resource utilization.
CO4: apply project planning, scheduling and monitoring techniques in engineering projects.
CO5: explain the concepts of marketing management and technology management in engineering organizations.

Mapping of COs 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. Management-A Global Perspective, Heinz Weihrich and Harold Koontz, McGraw Hill International Edition.
2. Industrial Engineering and Management -A New Perspective, Philip E. Hicks, McGraw Hill International Editions.
3. Essentials of Management, Andrew J. DuBrin, South-Western College Publishing.
4. Industrial Engineering and Management, O.P. Khanna and A. Sarup, Dhanpat Rai Publication Ltd.
5. Operations Management, William J Stevenson, McGraw Hill International Edition.

6. Project Management: A Systems Approach to Planning, Scheduling, and Controlling, Harold R. Kerzner, John Wiley and Sons.

SEMESTER -IV

CSE 0613 2237 Algorithms
3 hours in a week, 3 Credit

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

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/PLO	PLO1	PLO2	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.Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson.
2.Algorithms by Robert Sedgewick and Kevin Wayne.

CSE 0613 2238 Algorithms Sessional
3 hours in a week, 1.50 Credit
Laboratory work based on CSE 0613 2237.

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.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/PLO	PLO1	PLO2	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.Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson.
2.Algorithms by Robert Sedgewick and Kevin Wayne.

CSE 0613 2201 Digital Logic Design
3 hours in a week, 3 Credit
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	Design combinational and sequential circuits using logic gates.
CLO 2	Describe the principles of different types of counters and register.
CLO 3	Describe the operation and application of different memory and programmable logic device types.
CLO 4	Explain the basics of different logic families.

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

CSE 0613 2202 Digital Logic Design Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0613 2201**.

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	Implement combinational and sequential circuits using the logic gate.
CLO 2	Demonstrate proficiency in using laboratory tools to carry out the experiments.
CLO 3	Conduct experiments to correlate the theoretical knowledge.
CLO 4	Demonstrate the ability to work as a team.
CLO 5	Interface different IC using their specification sheet.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X			X					
CLO2				X	X				
CLO3			X						
CLO4									X
CLO 5					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

CSE 0714 2279 Computer Architecture
3 hours in a week, 3.00 Credit

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	Identify the fundamental technologiesincorporated in computer architectures.
CLO 2	Elevate the memory management technologies.
CLO 3	Implement pipelining mechanism and parallel computing in to the processor.
CLO 4	Improve I/O performance.

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

BUS 0411 2205 Accounting Fundamentals for IT Professionals
3 hours in a week, 3.00 Credit

Rationale

Accounting fundamentals for IT professionals is designed primarily to assist non-financial managers, investors and creditors in deciding where to place their scarce investment resources. They need sufficient knowledge of accounting to interpret a company's financial statements and make key business decisions. This course will help them manage their finances, understand profitability, and communicate effectively with investors. It will help them to develop effective software solutions: Creating accounting software, enterprise resource planning (ERP) systems, or financial analytical tools requires understanding the underlying accounting processes.

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: Define Accounting. Who uses Accounting Data? Types of Accounting, Accounting Concepts & Conventions, GAAP, Basic Accounting Equation: Assets, Liabilities, equity, Transaction Analysis, Summary of Transactions, Business problem and solution.

The Recording Process: Journal: Debits and Credits, Golden Rules of Accounting, rules of double entry, Summary Illustration of Journalizing, Summary Illustration of Posting from Journal to Ledger, T – Accounts, Preparation of a Trial Balance, and Limitations of a Trial Balance. (Discuss strengths and weaknesses and guide better performance in the future).

Adjusting the Accounts: The basics of adjusting entries, adjusting journals, the adjusted Trial balance, closing entries, and business problems.

Completing the Accounting Cycle: Worksheet, Summary of the Accounting Cycle, and Practice on Business Problem

Financial Statements: Income Statement, Balance Sheet, and Owner's Equity Statement (Services & Merchandise Concern)- Problem Solution from Annual Reports on Different Business Organizations.

Financial Statement Analysis: Basics of Financial Statement Analysis, Horizontal Analysis, Vertical Analysis, Ratio Analysis, Liquidity Ratio, Profitability Ratios, Solvency Ratios.

Cost-Volume-Profit: Cost Behavior Analysis, Variable Costs, Fixed Costs, Relevant Range, Mixed Costs, Importance of Identifying Variable and Fixed. Cost-Volume-Profit Analysis, Basic Components, CVP Income Statement, Break-even Analysis, Target Net Income, Margin of Safety. Variable Costing, Effects of Variable Costing on Income, Rationale for Variable Costing.

Managerial Accounting: Managerial Accounting Basics, Comparing Managerial and Financial Accounting, Management Functions, Organizational Structure, Business Ethics. Managerial Cost Manufacturing Costs, Product versus Period Costs, The Value Chain, Technological Change, Just-in-Time Inventory Methods, Quality, Activity-Based Costing, Theory of Constraints, Balanced Scorecard, Budgeting, cost of capital, capital budgeting.

Computerized Accounting Systems and Emerging Technologies: Overview of accounting software (e.g., Tally, QuickBooks, local solutions), Role of databases in accounting systems. Introduction to ERP systems and their accounting modules, Brief discussion on emerging trends: Cloud accounting, blockchain in accounting, AI's impact on financial reporting, Cybersecurity considerations in financial data.

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

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
CLO6	X						X		X
CLO7	X						X		X

Recommended Books:

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

MAT0541 2204 Complex Variables, Laplace's Transforms and Fourier Series
3 hours in a week, 3.00 Credit
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

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/PLO	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9

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

CSE 0610 2250 : Project Work II
3 hours in a week, 1.50 Credit
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	PLO1	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X		X	X	X	X			X
CLO2	X		X	X					X
CLO3		X	X	X					X
CLO4		X	X	X	X	X			X

CSE 0610 2290 Viva Voce
2 hours in a week, 1.00 Credit

Rationale:
This course objects to prepare students for their upcoming real-life interviews based on Departmental Subjects of Semester 3 to Semester 4

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	PLO2	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9

CLO1	X								X
CLO2	X								X
CLO3	X								X

SEMESTER -V

CSE 0612 3163 Database Management System
3 hours in a week, 3.00 Credit

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.

CSE 0612 3164 Database Management System Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0612 3163**.

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	Demonstrate the knowledge in projects with a commercial relational database system (Oracle) and design a team-based project.
CLO 2	Utilize the database design principles, SQL and PL SQL.
CLO 3	Demonstrate the relational database theory and be able to write relational algebra expressions for queries.

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.

. **CSE 0619 3193 Artificial Intelligence**
3 hours in a week, 3.00 Credit

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.

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

Textbook

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

CSE 0619 3194 Artificial Intelligence Sessional
3 hours in alternate week, 1.50 Credit

Laboratory works based on 0619 3193.

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

CSE 0612 3159 Web Technologies
2 hours in a week, 2.00 Credit

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

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	

CSE 0612 3160 Web Technologies Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0612 3159**.

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	

CSE 0714 3177 Microprocessors and Microcontrollers
3 hours in a week, 3.00 Credit

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 peripheral devices

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. Introduction to 8-bit, 16-bit, and 32-bit microprocessors: architecture, addressing modes, instruction set, interrupts, multi-tasking and virtual memory; Memory interface; Bus interface; Arithmetic co-processor; Microcontrollers; Integrating microprocessor with interfacing chips.

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

CSE 0714 3178 Microprocessors and Microcontrollers Sessional
3 hours in a week, 1.50 Credit

Laboratory works based on CSE **0714 3178**.

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

CSE 0714 3171 Data Communication
3 hours in a week, 3.00 Credit

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.
Signal and random processes; Review of ,Fourier Transform; Hilbert Transform, continuous wave modulation: AM, PM, FM; Sampling theorem; Pulse modulation: -PAM, PDM, PPM, PCM, companding, delta modulation, differential PCM; Multiple access techniques: TDM, FDM; Digital modulation: ASK, PSK, BPSK, QPSK; FSK, MSK, constellation, bit error rate (BER); Noise; Echo cancellation; Intersymbol Interference; Concept of channel coding and capacity.

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

CLO 1	Explain fundamental concepts of data communication and its components, different types of network, protocol layering and responsibilities of different layers for different protocols
CLO 2	Percept the digital and analogue representations of signals and analyze the mechanism of encoding schemas
CLO 3	Analyze resource sharing techniques called multiplexing for transmitting multiple signals through a single channel and identify techniques of different types of multiplexing.
CLO 4	Identify and analyze principles of security, performance and reliability of different networks.

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

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

CSE 0714 3172 Data Communication Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0714 3171**.

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

SEMESTER -VI

CSE 0619 3293 Machine Learning
3 hours in a week, 3.00 Credit

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 methods like principle component analysis, dimensionality reduction, feature extraction to prepare and manipulate data to make them suitable for ML approaches.
CLO 4	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.

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

CSE 0619 3294: Machine Learning Sessional
3 hours in a week, 1.50 Credit

Laboratory works based on CSE **0619 3294**.

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	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 methods like principle component analysis, dimensionality reduction, feature extraction to prepare and manipulate data to make them suitable for ML approaches.
CLO 4	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.

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

CSE 0613 3235 Operating System
3 hours in a week, 3.00 Credit
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	PL0 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)

CSE 0613 3236: Operating System Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0613 3235**.

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	PL0 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

CSE 0714 3275 Digital Signal Processing
3 hours in a week, 3.00 Credit

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.

CSE 0714 3276: Digital Signal Processing Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0714 3275**.

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.

CSE 0613 3241: Software Engineering and Design Patterns

3 hours in a week, 3.00 Credit

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-

CO1	Execute common software engineering processes and well-known practices.
CO2	Apply basic design principles and design pattern to write modular codes
CO3	Understand modern software verification and validation practices
CO4	Exercise project management process while developing software.

Mapping of Course Learning Outcomes to Program Learning Outcomes

CLO/ PLO	PLO1	PL 02	PLO3	PLO4	PLO5	PLO6	PLO7	PLO8	PLO9
CLO1	X								
CLO2			X						
CLO3	X								
CLO4									X

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

CSE 0613 3242: Software Engineering and Design Patterns Sessional Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0613 3242**.
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

CSE 0611 3230: Technical Writing And Presentation
3 hours in a week, 1.50 Credit

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

CSE 0610 3250 Project Work III
4 hours in a week, 2 Credit

Term project of making software on some practical problems with sound software engineering practices.
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 SEC 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	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			

CSE 0610 3290 Viva Voce
2 hours in a week, 1.00 Credit

Rationale:

This course objects to prepare students for their upcoming real-life interviews based on Departmental Subjects of Semester 5 to Semester 6.

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

SEMESTER -VII

CSE 0610 4122 Thesis I/CSE 0610 4144 Project I
4 hours in a week, 2.00 Credit

Study of problems in the field of Computer Science and Engineering.

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	

CSE 0612 4161: CLOUD COMPUTING

3 hours in a week, 3.00 Credit

Rationale:

Cloud Computing has become the dominant platform for building and delivering modern digital services by providing scalable computing, storage, networking, and managed services over the internet. It enables organizations to deploy applications faster, scale on demand, improve reliability, and optimize cost through pay-as-you-go models. This course provides undergraduate students with essential theoretical foundations of cloud concepts, service and deployment models, virtualization and containerization basics, cloud storage and networking, security and governance, cloud-native architecture principles, and emerging paradigms such as serverless computing—preparing students to analyze requirements and design secure, reliable, and cost-aware cloud solutions for real-world applications.

Objectives:

By the end of this course, students will be able to:

- Understand core concepts of cloud computing, including architecture, characteristics, and service/deployment models.
- Analyze virtualization and containerization concepts used to deliver cloud services.
- Apply concepts of storage, networking, and data services to propose scalable cloud architectures.
- Interpret cloud security, governance, and compliance requirements using appropriate control mechanisms.
- Evaluate cloud solution designs considering reliability, performance, and cost trade-offs.

Course Contents:

Cloud fundamentals: Evolution of cloud computing, characteristics of cloud, cloud architecture and roles, benefits and limitations, cloud economics (CapEx vs OpEx), overview of managed services and shared responsibility model. **Cloud service and deployment models**: IaaS, PaaS, SaaS; overview of FaaS/serverless; public, private, hybrid, and community cloud; multi-cloud concepts; cloud reference architectures and design patterns. **Virtualization and containerization fundamentals**: Hypervisors and virtual machines, VM lifecycle, images and snapshots, resource provisioning and elasticity concepts; containers, container images and registries, container lifecycle; container orchestration overview (Docker Swarm). **Cloud storage and data services**: Object, block, and file storage; durability and availability concepts; replication and consistency overview; managed relational and NoSQL databases; backup and data lifecycle basics. **Cloud networking**: Virtual networks (VPC/VNet), subnets, routing, NAT, and gateways; load balancing and autoscaling concepts; DNS and CDN fundamentals; network security groups/firewall concepts; private connectivity basics. **Cloud security and governance**: IAM concepts (RBAC, least privilege), authentication and authorization, encryption at rest and in transit, key management concepts, security monitoring and audit logs, governance policies, compliance overview, and data privacy basics. **Cloud-native architecture and reliability**: Stateless vs stateful services, microservices overview, API gateway concept, observability overview (logs, metrics, traces), high availability, fault tolerance, disaster recovery concepts (RPO/RTO). **Serverless and case studies**: Event-driven architecture, serverless benefits and constraints, queues/event buses overview, representative case studies (three-tier web application, data processing pipeline, IoT/edge overview), sustainability and ethical considerations in cloud usage.

Course Learning Outcomes:

CO No.	**CO Statement**
	After successful completion of the course, students will be able to:
CLO1	Explain fundamental concepts of cloud computing, including characteristics, architecture, service/deployment models, and shared responsibility
CO2	Analyze virtualization and containerization concepts and justify compute choices for different workload requirements
CO3	Apply cloud storage, database, and networking concepts to design a multi-tier cloud architecture.
CO4	Evaluate cloud security and governance controls (IAM, encryption, auditing) for given threat scenarios.
CO5	Create a complete cloud solution proposal, including reliability and cost reasoning for a defined problem.

Mapping of Course Outcomes to Program Outcomes:

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	X	X				X		X				X
CO2	X	X	X		X						X	
CO3	X	X	X	X	X				X	X	X	
CO4	X	X	X	X	X	X	X	X		X		X
CO5	X	X	X	X	X	X	X	X	X	X	X	X

Textbook

1. Rajkumar Buyya, Christian Vecchiola, S. Thamarai Selvi, *Mastering Cloud Computing*, McGraw-Hill.
2. Thomas Erl, Ricardo Puttini, Zaigham Mahmood, *Cloud Computing: Concepts, Technology & Architecture*, Pearson.

CSE 0612 4162: CLOUD COMPUTING SESSIONAL
3 hours in a week, 1.50 Credit

Rationale:

The Cloud Computing Lab complements the theory course by providing hands-on experience in provisioning and managing cloud resources, deploying applications, and practicing core cloud concepts such as virtualization, storage, networking, and security. Through guided experiments and a mini-project, students learn to translate architectural designs into working prototypes, observe system behavior using monitoring/logging tools, and apply best practices for secure and reliable cloud deployments. This practical exposure prepares students to work with modern cloud environments and develop real-world skills for building scalable and cost-aware cloud-based solutions.

Objectives:

By the end of this course, students will be able to:

- Implement basic cloud resource provisioning and management tasks for compute, storage, and networking.
- Apply containerization techniques to package and run applications.
- Configure access control and security settings using IAM principles and encryption options.
- Interpret monitoring and logging outputs to evaluate resource usage and application behavior.
- Develop and demonstrate an integrated cloud solution through a mini-project.

Course Contents:

Setup & fundamentals: Lab orientation, cloud console/CLI overview, regions/zones concept, resource naming/tagging, basic billing/free-tier awareness; environment setup for local tools (SSH, Docker, SDK/CLI). **Virtual machines and compute**: Provision VM instance, remote access (SSH), install and run a simple web service, snapshots/images, basic scaling concepts demonstration. **Storage services**: Object storage operations (bucket creation, upload/download, permissions), lifecycle policy demonstration; comparison of object vs block vs file storage (scenario-based). **Container fundamentals**: Create Dockerfile, build and run container image, port mapping, environment variables; container registry operations (tag, push, pull) using a registry. **Cloud networking**: Create a virtual network (VPC/VNet), public and private subnets, route tables, security group/firewall rules, NAT concept demonstration; deploy a two-tier setup (web + DB) with restricted access. **Load balancing and availability**: Configure a simple load balancer (or simulated), deploy multiple app instances, basic health checks, and a high availability demonstration. **Managed database services**: Provision managed relational/NoSQL database (or equivalent), connect application, perform CRUD operations, backup/restore demonstration. **IAM and security**: Create users/roles/policies, least privilege exercise, access testing (allowed/denied), enabling MFA concept discussion; encryption settings demonstration (at rest/in transit). **Monitoring and logging**: Enable logs and metrics, create a dashboard, basic alert rule (if available), and analyze performance/resource usage. Serverless mini-task (optional): Create a function triggered by HTTP or storage event, integrate with storage/logging. **Mini project**: Real-world problem selection, architecture diagram, implementation/prototype with compute + storage + network + security, basic monitoring, cost estimate summary, final report, and presentation. **Infrastructure as Code (IaC):** Provision infrastructure using declarative configuration, manage resources, execution planning (plan/apply), state management, configuration management basics, reusable scripts, and automation practices. **Configuration management & automation:** Automate server configuration using playbooks, inventory management, role-based configurations, idempotency concepts, and system provisioning automation. **Monitoring and logging:** Enable system and application monitoring, collect metrics, configure dashboards, define alert rules, centralized logging, log aggregation, and performance analysis. **DevOps security practices:** Secure pipelines, manage secrets and credentials, implement access control, basic vulnerability awareness, and secure configuration practices (DevSecOps concepts).

Capstone mini project: Real-world problem selection, design of an end-to-end DevOps pipeline, implementation including version control, CI/CD, containerization, infrastructure automation, monitoring integration, testing and optimization, final demonstration, report, and presentation.

Course Learning Outcomes:

CO No.	CO Statement
	After successful completion of the course, students will be able to:
CO1	Explain fundamental cloud concepts, including VM provisioning and resource management principles.
CO2	Provision and manage virtual machines and cloud resources in a guided environment.
CO3	Configure and use storage and database services to store and retrieve application data.

CO4	Implement secure cloud networking and IAM policies using subnets and controlled access mechanisms.
CO5	Design and develop an integrated cloud-based solution using compute, storage, networking, and monitoring components.

Mapping of Course Outcomes to Program Outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CLO1	X	X	X		X							X
CO2	X	X	X		X					X		X
CLO3	X	X	X	X	X				X	X	X	
CLO4	X	X	X	X	X	X	X	X		X		X
CLO5	X	X	X	X	X	X	X	X	X	X	X	X

Textbook

1. AWS Academy Cloud Foundation Course
2. AWS Academy Cloud Developing Course
3. AWS Academy Cloud Architecture Course

Tools

1. Cloud platform free tier/sandbox (AWS/Azure/GCP or university private cloud)
2. Docker
3. CLI/SDK tools (provider CLI)

CSE 0613 2131: Computer Graphics
3 hours in a week, 3.00 Credit

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.

CSE 0613 2132: Computer Graphics Sessional
3 hours in a week, 1.50 Credit

Laboratory works based on CSE **0613 2131**.

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,

Computer Graphics: A Programming Approach: Steven and Harrington

CSE 0612 4165: Computer Networks
3 hours in a week, 3.00 Credit

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	**Explain**the architecture of a computer network and **describe** how each device in a network communicates with each other
CLO 2	**Identify**the basic network protocols in each layer of a TCP/IP stack and **compare** with its counterpart OSI layer
CLO 3	**Compare** different LAN technologies and their basic principles
CLO 4	**Explain** the use of subnetting and use the technique to divide a large network into smaller logical subnetworks

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

CSE 0612 4166: Computer Networks Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0612 4165**.

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

OPTIONAL COURSES

CSE 0613 4167 **Simulation and Modeling**
3 hours in a week, 3.00 Credit

Simulation modeling basics: systems, models and simulation; Classification *of* simulation models; Steps in a simulation study; Concepts in discrete-event simulation: event-scheduling vs. process interaction approaches, time-advance mechanism, organization of a discrete-event simulation model; Continuous simulation models; Combined discreet-continuous models; Monte Carlo simulation; Simulation *of* queuing systems.
Building valid and credible simulation models: validation principles and techniques, statistical procedures for comparing real-world observations and simulated outputs, input modeling; Generating random numbers and random variates; Output analysis.
Simulation languages; Analysis and modeling *of* some practical systems.

CSE 0613 4168 **Simulation and Modeling Sessional**
3 hours in a week, 1.50 Credit
Laboratory works based on CSE 710.

OR

CSE 0619 4193 (3 credits) **DEEP LEARNING**
3 hours in a week, 3.00 Credit
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.

CSE 0619 4194 (1.5 credits**)** DEEP LEARNING LAB
3 Hours/Week
OR

CSE 0612 4165 (3 Credits) Compiler Design
3 hours in a week, 3.00 Credit

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.

CSE 0612 4166 (1.5 Credits) **Compiler Design Sessional**
3 hours in a week, 1.50 Credit

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

OR

CSE 0613 4171 **(3 Credits) Introduction to Quantum Computing**

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).

CSE 0613 4172 (1.5 Credits) Introduction to Quantum Computing LAB

3 Hours/Week

Laboratory works based on CSE 445

CSE 0714 4173 (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

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

CSE 0714 4174 (1.5 Credits) **INTRODUCTION TO INTERNET OF THINGS LAB**

3 Hours/Week

Laboratory works based on **CSE 0714 41**

SEMESTER -VIII

CSE 0610 4266 Thesis II/ CSE 0610 4288 Project II
8 hours in a week, 4.00 Credit

Study of problems in the field of Computer Science and Engineering.

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	

CSE 0613 4247 Cryptography and Network Security
3 hours in a week, 3.00 Credit

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	To get a basic understanding of principles and practice of cryptography and network security
CLO 2	Use of cryptographic algorithms in network protocols and network applications.
CLO 3	To be able to evaluate security of systems with respect to appropriate use of security services and mechanisms;
CLO 4	To be able to design basic security architectures through selection and integration of relevant security components.

Mapping of Course Learning Outcomes to Program Learning Outcomes

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

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

CSE 0613 4248 Cryptography and Network Security Sessonal
3 hours in a week, 1.50 Credit
Laboratory works based on **0613 4247**.

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 0613 4248**. 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

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					

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

CSE 0610 4290 Viva Voce
2 hours in a week, 1.00 Credit

Rationale:
This course objects to prepare students for their upcoming real life interviews based on Departmental Subjects of Semester 1 to Semester 8.

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

CSE 0613 4231 Digital Image Processing
3 hours in a week, 3.00 Credit
Introduction; Digitization of images and its properties; Data structures for image analysis; Image processing; Segmentation: detection of discontinuities, edge linking and boundary detection, thresholding, region oriented segmentation, use of motion in segmentation; Image transforms: Z-transform, 2D Fourier transform, discrete cosine transform, Hadamard transform, Walsh transform, Slant transform; Image compression: run-length coding, transform coding, standards.

CSE 0613 4232 Digital Image Processing Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on **0613 4231**

OR

CSE 0714 4271 Wireless & Mobile Communication
3 hours in a week, 3.00 Credit

Cellular concepts: frequency reuse, handoff strategies, interference and system capacity, grade of service, improving capacity and coverage, call blocking probability; Propagation effects: outdoor propagation models, indoor propagation models, power control, Doppler's effect, small and large scale fades; Wireless LAN Technology; IEEE 802.11: standard, protocol architecture, physical layer and media access control; Mobile IP; Wireless Application Protocol; IEEE 802.16 Broadband Wireless Access; Brief review of 2^{d} and $3^{`d}$ generation wireless: GSM, GPRS, CDMA; Cordless system; Wireless local loop; Bluetooth: overview and base band specifications.

CSE 0714 4272 Wireless & Mobile Communication Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0714 4271**.

OR

CSE 0613 4237 Advanced Algorithm Engineering
3 hours in a week, 3.00 Credit
Computational complexity, Parameterized complexity, Algorithms for combinatorial optimization, practical computing and heuristics, Approximation algorithms, LP based approximation algorithms, randomized algorithms, Experimental algorithmic, Algorithms in state of-the-art fields like Bioinformatics, Grid Computing, VLSI design etc.

CSE 0613 4238 Advanced Algorithm Engineering Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0613 4237**.

OR

CSE 0688 4297 Bioinformatics
3 hours in a week, 3.00 Credit
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.

CSE 0688 4298 Bioinformatics Sessional

Credits: 1.5 **Contact hours:** 3 hours/week

Laboratory works based on CSE **0688 4297**.

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 by PavelPevzner and Ron Shami

OR

CSE 0714 4277 VLSI Design
3 hours in a week, 3.00 Credit

VLSI design methodology: top-down design approach, technology trends and design automation algorithms; Introduction to CMOS inverters and basic gates; Brief overview of CMOS fabrication process: layout and design rules; Basic CMOS circuit characteristics and performance estimation; Buffer circuit design; Complex CMOS gates, CMOS building blocks: adder, multiplier; data path and memory structures.
Hardware modeling: hardware modeling languages, logic networks, state diagrams, data-flow and sequencing graphs, behavioral . optimization.
Architectural Synthesis: circuit specification, strategies for architectural optimization, data-path synthesis, control unit synthesis and synthesis of pipelined circuits.
ASIC design using FPGA and PLDs.

CSE 0714 4278 VLSI Design Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0714 4277**.

OR

CSE 0613 4243 Natural Language Processing
3 hours in a week, 3.00 Credit
Introduction; Word Vectors, Word Modeling: Word Senses, Word Window Classification, Neural Networks, and Matrix Calculus, Automata and Linguistics, Statistical Approaches and Part of Speech Tagging; Linguistics and Grammars; Dependency Parsing, 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; Language Models and Recurrent Neural Networks, Machine Translation, ConvNets for NLP, Subword Models, Contextual Word Representations and Pretraining, Generative Models, Natural Language Generation, Coreference Resolution, Tree Recursive Neural Networks, Constituency Parsing, and Sentiment, Future Deep Learning and NLP.

CSE 0613 4244 Natural Language Processing Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0613 4243**.

CSE 0612 4267 Distributed and Parallel Computing
3 hours in a week, 3.00 Credit

Introduction to Parallel and Distributed Systems: Architecture, Challenges, principle and paradigm, Middleware: Introduction to Erlang, Communication: Synchronous and asynchronous communication abstraction and model, message passing and shared memory. Replication & Consistency: Control replication, data replication, consistency model and protocols. Distributed Shared Memory: Design issue, Implementation issue, consistency issue, Shared Memory model, MPI, LINDA, ORCA, case study: Trademark, JACKAL, Distributed Objects: Introduction, remote objects, CORBA, Distributed Shared object, Globe. Synchronization & Coordination: Distributed algorithms, time and clocks, Local state, Global State, consistency protocols, coordination elections, distributed transactions management. Fault Tolerance: Failure model, Faults, Process Resilience, reliable communication, Recovery, Checkpoints and checkpoint algorithms, Rollback recovery algorithms, Security: Threats and attacks, policy and mechanism, Design issue, design of cryptographic algorithms, cryptographic protocols, Key distribution, authentication, secure communication, auditing. Naming: Basic concept, Naming Services, DNS, Attribute based naming, X.500 and LDAP, Distributed File Systems: Client perspective, Server perspective, NFS, Coda, Google File System (GFS), Parallel Programming: parallel computing, parallel programming structure, Planet Lab, Grid: Grid model, Grid Middleware, Globus toolkit, Planet Lab Overview.

CSE 0612 4268 Distributed and Parallel Computing Sessional
3 hours in a week, 1.50 Credit
Laboratory works based on CSE **0612 4267**.

CSE 0612 4269 Generative AI and Large Language Models
3 hours in a week, 3.00 Credit

Rationale:
Generative AI and Large Language Models (GenAI/LLM) is an advanced course in the AI/ML track of Computer Science and Engineering. Building on prerequisite knowledge of deep learning, transformer architecture, and sequence modeling, this course addresses higher-order engineering competencies demanded by modern AI practice. Topics include advanced LLM architecture internals, prompt engineering, contextual embeddings, retrieval-augmented generation, parameter-efficient fine-tuning, autonomous agent design, evaluation methodology, model alignment and safety, mechanistic interpretability, and production deployment. The course prepares students to design, implement, evaluate, and deploy large language model systems in real-world settings.

Objectives:
By the end of this course, students will be able to:

- Explain advanced LLM architecture internals including tokenization, scaling laws, positional encoding schemes, grouped-query attention, mixture-of-experts, and the pre-training to alignment pipeline.
- Design and evaluate systematic prompt engineering strategies including chain-of-thought, self-consistency, tree-of-thoughts, and automated prompt optimization.
- Construct end-to-end retrieval-augmented generation systems using advanced retrieval, reranking, and evaluation frameworks.
- Design agentic AI systems with multi-agent collaboration and apply alignment, safety, and governance frameworks.
- Evaluate LLM systems using interpretability methods and inference-time scaling; architect production-grade deployments with LLMOps practices.

Course Contents:
LLM Landscape and Advanced Architecture: Compressed GenAI taxonomy; major LLM families (GPT-4o, Claude, Gemini, LLaMA, Mistral, DeepSeek); closed-source vs. open-weight trade-offs; grouped-query attention (GQA) and multi-query attention (MQA); rotary positional embedding (RoPE) and ALiBi; sliding window and sparse attention; mixture-of-experts (MoE) architecture; tokenization in depth: BPE, SentencePiece, tiktoken. **LLM Training Pipeline and Alignment:** Pre-training data curation; scaling laws and Chinchilla compute-optimal insights; emergent abilities; supervised fine-tuning (SFT) and instruction tuning; reinforcement learning from human feedback (RLHF): reward model training, PPO, reward hacking; direct preference optimization (DPO); Constitutional AI and RLAIF; synthetic data generation for LLM training. **Prompt Engineering:** Prompt anatomy; zero-shot, few-shot, and one-shot prompting; structured output formats; prompt injection attacks and defenses; chain-of-thought (CoT), self-consistency, tree-of-thoughts (ToT), ReAct pattern; meta-prompting; automatic prompt optimization (APE, OPRO); evaluation pipelines using Promptfoo. **Contextual Embeddings and Semantic Representations:** Contextual embeddings: BERT-family, Sentence-BERT, E5, BGE; dense vs. sparse retrieval; bi-encoder and cross-encoder architectures; contrastive learning and SimCSE; Matryoshka representation learning; embedding evaluation with the MTEB benchmark. **Retrieval-Augmented Generation:** RAG pipeline; document loading and chunking strategies; vector databases (FAISS, ChromaDB, Pinecone, Weaviate); hybrid search (dense + BM25); cross-encoder reranking; advanced RAG: HyDE, multi-query retrieval, FLARE; deep research agents; RAGAS evaluation framework; RAG vs. fine-tuning decision framework. **Fine-Tuning and Synthetic Data:** Full fine-tuning vs. PEFT; SFT and instruction tuning; catastrophic forgetting; LoRA and QLoRA; prefix tuning; adapter methods; synthetic data generation via self-instruct; cost-benefit evaluation. **LLM Evaluation and Inference-Time Reasoning:** Intrinsic and extrinsic metrics; human evaluation protocols; LLM-as-judge; industry benchmarks (MMLU, HumanEval, MATH, GPQA); inference-time scaling: extended chain-of-thought, process reward models (PRM), outcome reward models (ORM), Best-of-N sampling. **AI Agents and Multi-Agent Systems:** Agent taxonomy; tool use and function calling; memory architectures; LangChain agents; LangGraph stateful graphs; multi-agent collaboration: AutoGen, CrewAI; deep research agents; agent observability with LangSmith. **Alignment, Safety, Ethics, and Governance:** Hallucination types and mitigation; bias and fairness; toxicity detection; prompt injection and jailbreaking; LLM security threat taxonomy; guardrails frameworks; scalable oversight; EU AI Act; NIST AI RMF. **Mechanistic Interpretability:** Probing classifiers; attention visualization; activation patching and causal tracing; superposition hypothesis; circuits and induction heads; representation engineering; practical tools: TransformerLens, Baukit. **Deployment, LLMOps, and Production Systems:** Quantization: GPTQ, AWQ; KV cache optimization; paged attention; inference serving: vLLM, TGI; speculative decoding; semantic caching; deployment: Docker, FastAPI, serverless; observability; LLMOps: experiment tracking, prompt versioning, CI/CD evaluation. **Code LLMs, Multimodal AI, and Emerging Frontiers:** Code LLMs: CodeLlama, DeepSeek-Coder, SWE-bench, fill-in-the-middle training; vision-language models; multimodal evaluation; long-context modeling; MoE scaling; open-source ecosystem trends.

Course Learning Outcomes:

CO No.	CO Statement
	After successful completion of the course, students will be able to:
CO1	Explain advanced LLM architecture internals including tokenization, scaling laws, positional encoding schemes, grouped-query attention, mixture-of-experts, and the full pre-training to alignment pipeline.

CO2	Design and evaluate systematic prompt engineering strategies including chain-of-thought, self-consistency, tree-of-thoughts, ReAct, and automated prompt optimization pipelines.
CO3	Construct end-to-end retrieval-augmented generation systems incorporating advanced retrieval, hybrid search, reranking, fine-tuning integration, and evaluation using the RAGAS framework.
CO4	Design agentic AI systems incorporating tool use, memory, multi-step reasoning, and multi-agent collaboration; apply alignment, safety, and governance frameworks to LLM deployments.
CO5	Evaluate LLM systems using intrinsic and extrinsic metrics, inference-time scaling, and interpretability methods; architect production-grade deployments with LLMOps practices.

Mapping of Course Outcomes to Program Outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1									
CO2	3	2	2	1								
CO3	2	3	3	2	1							
CO4	2	3	2			2	2					
CO5	2	2	2						1	2		1

Textbook

1. Vaswani, A. et al. (2017). Attention Is All You Need. NeurIPS. [Foundational architecture]
2. Touvron, H. et al. (2023). LLaMA 2: Open Foundation and Fine-Tuned Chat Models. Meta AI.
2. Lewis, P. et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS.
3. Hu, E. et al. (2021). LoRA: Low-Rank Adaptation of Large Language Models. ICLR 2022.
4. Wei, J. et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS.
5. Bai, Y. et al. (2022). Constitutional AI: Harmlessness from AI Feedback. Anthropic.
6. Elhage, N. et al. (2022). Toy Models of Superposition. Anthropic Interpretability Research.

CSE 0612 4270 Generative AI and Large Language Models Sessional
3 hours in a week, 1.5 Credit

Rationale:
The Generative AI and Large Language Models Lab complements the theory course by enabling students to implement and experiment with LLM-based systems in practical settings. It bridges the gap between conceptual knowledge — including prompting strategies, retrieval-augmented generation, fine-tuning, agent design, and deployment — and hands-on implementation using modern tools and platforms. Through structured lab exercises and a capstone project, students gain skills in building, evaluating, and deploying production-ready generative AI applications, which are essential competencies in software engineering, data science, and applied AI.

Objectives:
By the end of this course, students will be able to:
- Implement multi-provider LLM API interactions and build prompt engineering libraries using automated optimization and evaluation tools.
- Construct semantic search engines and complete RAG pipelines from ingestion to evaluation and deployment.
- Execute parameter-efficient fine-tuning on custom instruction datasets and produce quantitative evaluation reports.
- Build multi-agent systems and conduct adversarial safety auditing with guardrail validation pipelines.
- Develop a complete GenAI application addressing a real-world problem with deployment, monitoring, and LLMOps practices.

Course Contents:
LLM API Fundamentals and Prompt Engineering: Multi-provider API integration (OpenAI, Anthropic, Google Gemini, Groq); token cost measurement and model comparison; API response handling; building prompt libraries; zero-shot, few-shot, and chain-of-thought implementations; automatic prompt optimization using APE/OPRO-style methods; evaluation with Promptfoo; systematic A/B testing pipelines. **Embeddings and Semantic Search:** Sentence-transformer embedding models (Sentence-BERT, E5, BGE); vector index construction with FAISS and ChromaDB; hybrid BM25 and dense retrieval; cross-encoder reranking; evaluation using NDCG and MTEB benchmarks; semantic search application development with Streamlit. **Retrieval-Augmented Generation and Fine-Tuning:** End-to-end RAG pipeline implementation using LangChain; document ingestion and chunking strategies; advanced RAG techniques: HyDE, FLARE, multi-

query retrieval; RAGAS evaluation for faithfulness, answer relevance, context precision, and context recall; deployment as a FastAPI service; synthetic instruction dataset generation using self-instruct methodology; QLoRA fine-tuning with Unsloth and HuggingFace PEFT; 4-bit quantization with bitsandbytes; experiment tracking with WandB; before-vs-after quantitative benchmarking. **LLM Evaluation and Inference-Time Reasoning:** Evaluation using RAGAS, BERTScore, and LLM-as-judge; industry benchmark evaluation (MATH benchmark); Best-of-N sampling implementation; process reward model (PRM) scoring; reasoning analysis and evaluation dashboard construction. **Multi-Agent Systems and Safety Auditing:** Stateful agent construction using LangGraph; multi-agent orchestration with AutoGen and CrewAI; tool use and function calling; persistent memory management; human-in-the-loop controls; observability and tracing with LangSmith; adversarial red-teaming exercises; guardrail pipeline implementation with Guardrails.ai; toxicity detection using Perspective API; bias auditing. **Mechanistic Interpretability:** Attention visualization using BertViz; probing classifier implementation; activation patching and causal tracing experiments using TransformerLens; induction head analysis; interpretability notebook and analysis report. **Deployment, LLMOps, and Capstone Project:** REST API development with FastAPI; containerization with Docker; semantic caching with GPTCache; monitoring and observability with Helicone; LLMOps experiment tracking with WandB; CI/CD-integrated evaluation pipeline design; real-world problem selection; full pipeline implementation covering data ingestion, LLM integration, evaluation, and deployment; GitHub-hosted codebase, technical report, and Demo Day presentation.

Course Learning Outcomes:

CO No.	CO Statement
	After successful completion of the course, students will be able to:
CO 1	Implement multi-provider LLM API interactions, construct prompt engineering libraries, and apply automated optimization and evaluation pipelines using standard tools.
CO 2	Build semantic search engines using sentence-transformer embeddings, vector databases, and hybrid retrieval with quantitative evaluation using NDCG and MTEB benchmarks.
CO 3	Construct a complete RAG pipeline from document ingestion through advanced retrieval techniques and RAGAS evaluation, and execute QLoRA fine-tuning on a custom synthetic instruction dataset.
CO 4	Build a multi-agent system using LangGraph and AutoGen with tool use, persistent memory, and human-in-the-loop controls; conduct adversarial red-teaming and implement guardrail validation pipelines.
CO 5	Design and develop a complete GenAI application addressing a real-world problem, deployed as a Dockerized REST API service with semantic caching, observability, and LLMOps experiment tracking.

Mapping of Course Outcomes to Program Outcomes:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2										1
CO2	3	3	2	1								
CO3	3	3	3	2								1
CO4	2	3	2			3		2				
CO5	2	3	2	2						2		1

Textbook

1. Vaswani, A. et al. (2017). Attention Is All You Need. NeurIPS.
2. Hu, E. et al. (2021). LoRA: Low-Rank Adaptation of Large Language Models. ICLR 2022.
3. Lewis, P. et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS.
4. Wei, J. et al. (2022). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS.
5. Bai, Y. et al. (2022). Constitutional AI: Harmlessness from AI Feedback. Anthropic.

Tools

1. Python 3.11+
2. OpenAI API, Anthropic API, Google Gemini API, Groq
3. LangChain, LangGraph, AutoGen, CrewAI
4. ChromaDB, FAISS, Pinecone
5. Unsloth, HuggingFace PEFT, bitsandbytes
6. RAGAS, Promptfoo, BERTScore, LangSmith
7. TransformerLens, BertViz, Baukit
8. FastAPI, Docker, Streamlit, GPTCache, Helicone, WandB
9. Guardrails.ai, NeMo Guardrails, Perspective API