Course Description- CSE
1st Semester

Course Name: Engineering Mathematics **Course Code: MA101**

L-T-P scheme: 3-1-0 **Credits: 4**

Prerequisite: Students should have basic knowledge of Algebra and calculus.

Objective: This course is aimed:

- To introduce the calculus of functions of two variables and applicability of derivatives and integrals of vector functions to Analytical geometry and physical problems.
- To make students aware of the basic mathematical concepts and methods which will help them in learning courses in engineering and Technology.

Learning Outcomes:

Course Outcome	Description
CO1	Understand the rank, eigen values, eigen vectors, diagonalization of matrix; compute inverse of matrix by Caley-Hamilton theorem.
CO2	Recognize consistent and inconsistent systems of linear equations by the row echelon form of the augmented matrix, and solve it by Gauss elimination method.
CO3	Interpret derivatives and integrals of multivariable functions geometrically and physically; implement multivariable calculus tools in engineering, science, optimization, and understand the architecture of surfaces in plane and space etc.
CO4	Know about piecewise continuous functions, Laplace transforms and its properties; use of Laplace transform and inverse transform for solving initial value problems.
CO5	Realize importance of line, surface and volume integrals, Gauss and Stokes theorems and apply the concepts of vector calculus in real life problems.
CO6	Formulate mathematical models in the form of ordinary differential equations and learn various techniques of getting solutions of linear differential equations of second order.

Course Contents:

Unit 1: Algebra of matrices, Determinants, Rank, Gauss elimination method, Eigen values and vectors. Quadratic forms.

Unit 2: Partial differentiation. Taylor's series. Maxima and minima. Jacobians**,** Double integrals,

Unit 3: Differential Equations with constants coefficients.

Unit 4: Gradient, divergence and curl. Line and surface integrals, Normal and tangent to a surface. Gauss and Stokes theorems, Equations to a line, plane, curve and surfaces.

Unit 5: Laplace transforms.

Methodology:

The course will be covered through lectures supported by tutorials. There shall be 3 Lectures per week where the teacher will explain the theory, give some examples supporting the theory and its applications. About 12 Tutorial Sheets covering whole of the syllabus shall be given. Difficulties and doubts shall be cleared in tutorials. Apart from the discussions on the topics covered in the lectures, assignments/ quizzes in the form of questions will also be given.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered upto Test-1
Test-2	25 Marks	Syllabus covered upto Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials, lecture slides and books on mathematics-1 will be available on the JUET server.

Books

1. Erwin Kreyszig: Advanced Engineering Mathematics, Wiley Publishers.
2. Lipshuts, S., Lipsom M.: Linear Algebra, 3rd Ed, Schaum series 2001.
3. B. V. Raman: Higher Engineering Mathematics, McGraw-Hill Publishers.
4. R.K. Jain, S.R.K. Iyenger: Advanced Engineering Mathematics, Narosa Publishing House, New Delhi.
5. Thomas, G.B., Finney, R.L.: Calculus and Analytical Geometry, 9th Ed., Addison Wesley,1996.
6. Grewal, B.S. : Higher Engineering Mathematics, Khanna Publishers Delhi.

Title of Course: Engineering Physics-I **Course Code: PH101**

L-T Scheme: 3-1-0 **Course Credits: 4**

Objective: Broadly, the study of Physics improves one's ability to think logically about the problems of science and technology and obtain their solutions. The present course is aimed to offer a broad aspect of those areas of Physics which are specifically required as an essential background to all engineering students for their studies in higher semesters. The course intends to impart sufficient scientific understanding of different phenomena associated with Special relativity, Modern Physics, Statistical physics, atomic physics, and lasers.

Course Outcomes:

Course Outcome	Description
CO1	Describe the limitations of Newton's laws and explain when special relativity become evant, Learn to Apply the principles of Special Relativity to an extended range of problems volving particle kinematics
CO2	Demonstrate the ability to explain the concepts related to the consequences of Special Relativity, the nature of space-time and related dynamic observables
CO3	Acquired a profound understanding of inadequacy of classical mechanics regarding phenomena related to microscopic level, Become well versed with the experimental developments, historical account and importance of probabilistic interpretation
CO4	Understand the basic quantum mechanical ideas and relevant mathematical framework, approach the solution of one dimensional time independent Schrodinger equation
CO5	Appreciate the importance of applying statistical ideas to explore thermodynamic variables, Developed ability to identify and apply appropriate statistical method for describing the assembly of microscopic particles, comprehend basic properties and working of Laser systems

Course Contents:

Unit-I (Theory of Special Relativity): Frames of reference, Galilean transformation, Michelson Morley Experiment, Postulates of special theory of relativity, time dilation and length contraction, twin paradox, Lorentz transformations, addition of velocities, Relativistic Doppler effect, Mass variation with velocity, Mass-energy relation.

Unit-II (Introduction to Modern Physics):

Quantization of Radiation, Black body radiation, Rayleigh-Jeans law, Planck's law of radiation Wien's law, Stefan's law, Photoelectric effect Compton scattering, Atomic spectra, Bohr model of hydrogen atom, Frank hertz experiment, Matter waves, de Broglie hypothesis, Davisson Germer experiment

Unit III Quantum Mechanics

Wave packets, phase and group velocity, Heisenberg's uncertainty principle, Schrödinger wave equation and its applications to the free particle in a box, potential barrier and Harmonic oscillator

Unit-IV (Statistical Mechanics): Maxwell-Boltzmann, Bose-Einstein and Fermi-Dirac distributions and their applications.

Unit- V Laser Physics & Applications

Fundamental ideas of stimulated and spontaneous emission, Einstein's coefficients, Principle and working of laser, Different types of lasers (He-Ne Laser, Ruby Laser, Semiconductor Laser), Applications of Lasers

Text Books and References:

1. A. Beiser, Perspectives of Modern Physics, Tata McGraw Hill.
2. J R Taylor, C D Zafiratos, M A Dubson, Modern Physics for Scientist &
1. Engineers, Pearson Education.
2. K Krane, Modern Physics, Wiley India
3. J Bernstein, P M Fishbane, S. Gasiorowicz, Modern Physics, Pearson
4. Education.
5. B. B. Laud, Laser and Non-Linear Optics, New Age International (P) Ltd.
6. R. Resnick, Relativity, New Age.

Title: English **Code: HS101**

L-T-P scheme: 2-1-0 **Credit:** 3

Prerequisite: None

Objective:

1. To enable understanding of basics of communication in Business environment.
2. To provide insight into structural aspect of communication in business.
3. To impart knowledge about communication theory and develop skills in oral and non verbal communication.
4. To improve skills as critical readers, thinkers, listener and writer.

Learning Outcomes:

Course Outcome	Description
CO1	Outline the basic concept of verbal/ nonverbal skills to understand the role of effective communication in personal & professional success.
CO2	Describe drawbacks in listening patterns and apply listening techniques for specific needs.
CO3	Develop the understanding to analyze, interpret and effectively summarize a variety of textual content
CO4	Discuss a given technical/non-technical topic in a group setting and arrive at generalizations/consensus.
CO5	Create effective presentations
CO6	Create professional and technical documents that are clear and adhering to all the necessary convention.

Course Content:

Unit-1: Concept and Nature of Communication : Definition of Communication, Process & Stages of Communication, Barriers to Communication, Channels of Communication.

Unit-2: Listening Skills: The listening process**,** Importance of listening, Purpose and types of listening**,** Hearing and listening, Listening with a purpose, Barriers to listening.

Unit-3: Speaking/Oral Skills: Importance of acquiring oral skills, Visual aids, Body Language, Delivery, Pronunciation, Use of connectives Organization of matter: Metadiscourse features, Textual organization, 7 C'S of effective communication , Improving vocabulary by learning Root words in English, Some foreign words, Reading comprehension, Some important synonyms and antonyms, commonly confused words, Etiquettes & grooming.

Unit-4: Reading Skills: Skimming and Scanning, Intensive and extensive reading, SQ3R Technique

Unit-5: Writing Skills: Business letters, Memo, Circulars, Notices, Report writing, resume writing, Agenda & Minutes writing, Tips on clear writing Translation- Hindi to English, Translation -English to Hindi.

Unit-6: Introduction to Modern Communication Media: Technology based communication tools, Committee types, Advantages, Conferences, Audio-video conferencing, Barriers and overcoming negative impact.

Unit-7: Public Speaking and Interviewing Strategies: Speech Preparation, Theory of group discussion, Participation in Group discussion, Oral presentation, Power point presentation ,Tips for successful job interview, Do's and don'ts while appearing for interview, Mock interview, Some interview questions, Telephonic interview tips, Resume writing

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1 & Unit-2
Test-2	25 Marks	Based on Unit-3,& Unit-4 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-5 to Unit-7 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Teaching Methodology:

The course will be taught with the aid of lectures, handouts, case studies, Task-based language learning, and comprehensive language learning through language lab.

Learning Resources:

Lecture slides and e-books on ENGLISH (will be added from time to time): Digital copy will be available on the JUET server.

Text Book:

1. K.K. Sinha- Business Communication (Galgotia Publications)

Reference Books:

1. R.C. Bhatia- Business Communication (Ane Books Pvt. Ltd.)
2. P.D. Chaturvedi – Business Communication (Pearson Education, 1st Edition 2006).
3. Lesikar RV & Pettit Jr. JD – Basic Business Communication: Theory & Application (Tata Mc Graw Hill, 10thEdition)
4. Wren & Martin, High School English Grammar & Composition – S. Chand & Co. Delhi.

5. Raman Meenakshi & Sharma Sangeeta, Technical Communication-Principles & Practice –O.U.P. New Delhi. 2007.
6. Mitra Barum K., Effective Technical Communication – O.U.P. New Delhi. 2006.
7. Better Your English- a Workbook for 1st year Students- Macmillan India, New Delhi.
8. Raymond Murphy,' Essential English Grammar', Cambridge University Press.

Title: Computer Programming **Code: CS101**
L-T-P scheme: 3-1-0 **Credit: 4**

Prerequisite: There is no prerequisite in this course; however, students having any prior experience of programming are desirable.

Objective:

1. To provide exposure to problem-solving through programming.
2. To provide students with understanding of code organization and functional hierarchical decomposition with using complex data types.

Learning Outcomes:

Course Outcome	Description
CO1	Makes students gain a broad perspective about the uses of computers in engineering industry.
CO2	Develops basic understanding of computers, the concept of algorithm and algorithmic thinking.
CO3	Develops the ability to analyze a problem, develop an algorithm to solve it.
CO4	Develops the use of the C programming language to implement various algorithms, and develops the basic concepts and terminology of programming in general.
CO5	Introduces the more advanced features of the C language

Course Content:

Unit-1: Introduction to Programming: Basic computer organization, operating system, editor, compiler, interpreter, loader, linker, program development. Variable naming, basic function naming, indentation, usage and significance of comments for readability and program maintainability. Types of errors, debugging, tracing/stepwise execution of program, watching variables values in memory. Constants, Variables and data Types Character Set, C tokens, Keywords and Identifiers, Constants, Variables, Data types, Declaration of Variables, assigning values to variables, typedef, and Defining symbolic constants. printf & scanf function.

Unit-2: Operators and Expression: Introduction, Arithmetic Operators, Relational Operators, Logical Operators, Assignment Operators, Increment and Decrement Operators, Conditional Operators, Special Operators, Evaluation of expressions, Precedence of arithmetic operators, Type conversions in expressions, Operator precedence and associativity.

Management Input and Output Operators: Introduction, reading a character, writing a character, formatted input, formatted output.

Unit-3: Decision Making Branching: Introduction, Decision making with IF statement, the IF-ELSE statement, nesting of IF-ELSE statement, ELSE-IF ladder, SWITCH statement, ternary operator, and the GOTO statement.

Looping: Introduction, the WHILE statement, the DO statement, The FOR statement, Break and Continue.

Unit-4: Array: Introduction, One-dimensional arrays, Two-dimensional arrays, arrays, Concept of Multidimensional arrays.

Handling of Character strings: Introduction, Declaring and initializing string variables, reading string from terminal, writing string to screen, String, Operations: String Copy, String Compare, String Concatenation and String Length (using predefined functions & without using them), Table of strings.

Unit-5: User-Defined Functions (UDF): Introduction, need for user-defined functions, the form of C function, elements of UDF, return values and their types, Calling a function, category of functions, Nesting of functions, Recursion, Functions with arrays, The scope and Lifetime of variables in functions, multi file program.

Structures and Unions: Introduction, Structure definition, declaring and initializing Structure variables, accessing Structure members, Copying & Comparison of structures, Arrays of structures, Arrays within structures, Structures within Structures, Structures and functions, Unions.

Unit-6: Pointers: Introduction, understanding pointers, Accessing the address of variable, Declaring and initializing pointers, accessing a variable through its pointer, Pointer expressions, Pointer increments and scale factor, Pointers and arrays, Pointers & character strings, Pointers & Functions, Function returning multiple values, Pointers and structures.

File Management in C and CONSOLE I/O: Introduction, Defining files and its Operations, Error handling during I/O operations, Random access files, Command line arguments. Types of files, File vs. Console, File structure, File attributes, Standard i/o, Formatted i/o, Sample programs.

Teaching Methodology:

This course is introduced to help students understand the discipline of programming. The programming language used to teach this course is C. Starting from the basic computer architecture, the student will slowly be exposed to program designing and later to programming fundamentals. The entire course is broken down into six separate units, from fundamentals of programming to some complex programming structures like pointers. This theory course is well complemented by a laboratory course under the name Software Development Fundamentals Lab in the same semester that helps a student learn with hand-on experience.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1 & Unit-2
Test-2	25 Marks	Based on Unit-3 & Unit-4 and around 20-30% from coverage till Test-1
Test-3	35 Marks	Based on Unit-5 to Unit-6 and around 30% from coverage till Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Software Development Fundamentals (will be added from time to time): Digital copy will be available on the JUET server.

Text Book:

[1] Programming in ANSI C by E. Balguruswamy, Tata Mc-Graw Hill.
[2] Programming With C, Schaum Series.

Reference Books/Material:

[1] The 'C' programming language by Kernighan and Ritchie, Prentice Hall
[2] Computer Programming in 'C' by V. Rajaraman, Prentice Hall
[3] Programming and Problem Solving by M. Sprankle, Pearson Education
[4] How to solve it by Computer by R.G. Dromey, Pearson Education

Web References:

[1] http://www2.its.strath.ac.uk/courses/c/
Notes on C programming by University of Strathclyde Computer Centre. This tutorial was awarded the NetGuide Gold Award during the 1990s.

[2] http://www.princeton.edu/~achaney/tmve/wiki100k/docs/C_%28programming_language%29.html
This site contains notes on C programming from Princeton University, USA.
These are very useful for students who are learning C as their first programming Language.

[3] http://www.stat.cmu.edu/~hseltman/Computer.html
Online reference material on Computers and Programming from Carnegie Mellon University, Pittsburgh, USA

[4] http://projecteuler.net/
Collection of mathematical problems which make you use your programming skills

Title: Engineering Physics Lab-I **Code: PH201**
L-T-P scheme: 0-0-2 **Credit: 1**

Learning Outcomes

Course Outcome	Description
CO1	Demonstrate ability to collect experimental data and understanding the working procedures within the precautionary limits
CO2	Acquired the ability to analyze the experimental data and related errors in a reflective, iterative and responsive way
CO3	Developed understanding of the basic concepts related to Modern Physics, Basic Solid State Physics and Optics
CO4	Acquired a first hand and independent experience of verifying Kirchoff's circuit laws and related concepts e.g. resistivity, measurement of resistance
CO5	Appreciate the importance of the laboratory work culture and ethics that is intended to impart features like regularity, continuity of self evaluation and honesty of reporting the data

List of Experiments

1. To study the variation of magnetic field along the axis of Helmholtz Galvanometer and to determine its reduction factor.
2. To determine the resistance per unit length of a Carey Foster's bridge and to obtain the specific resistance of a given wire.
3. To determine the wavelengths of spectral lines Red, Green and Violet of mercury using plane transmission grating.
4. To determine the specific rotation of cane sugar solution using Bi-quartz polarimeter.
5. To observe Newton's rings and to determine the wavelength of sodium light.

6. To study the CRO and function generator by producing the following waveforms.
    i. 10kHz, 8Vp-p(sine wave, square wave, triangular wave)
    ii. 4kHz, 6Vp-p(sine wave, square wave, triangular wave)
    iii. 10kHz, 8Vpeak(sine wave, square wave, triangular wave)
    iv. 4kHz, $6V_{peak}$(sine wave, square wave, triangular wave)
7. To verify the Kirchhoff's current law.
8. To verify the Kirchhoff's voltage law.

Title: Computer Programming Lab **Code: CS201**

L-T-P scheme: 0-0-4 **Credit: 2**

Prerequisite: Experience in programming is desirable.

Objective:

1. To provide exposure to problem-solving through programming.
2. To provide students with understanding of code organization and functional hierarchical decomposition with using complex data types.
3. To give the student hands-on experience with the concepts.

Learning Outcomes:

Course Outcome	**Description**
CO1	Makes students gain a broad perspective about the uses of computers in engineering industry.
CO2	Develops basic understanding of computers, the concept of algorithm and algorithmic thinking.
CO3	Develops the ability to analyze a problem, develop an algorithm to solve it.
CO4	Develops the use of the C programming language to implement various algorithms, and develops the basic concepts and terminology of programming in general.
CO5	Introduces the more advanced features of the C language

Course Content:

The following assignments will be carried out in synchronization with the theory classes.

Unit-1: Introduction to programming Environment (Linux commands, editing tools such as vi editor, sample program entry, compilation and execution). Development of programs using multiple arithmetic and logical operators. Programs for Roots of quadratic equation, conversion of units etc.

Unit-II: Programs using simple control statements such as if else, while, do while etc. Making a program for a calculator for example. Extracting the digits of an integer, reversing digits, finding sum of digits etc.

Unit-III: Programs using For loop, switch statement etc. For example, Finding average of numbers, printing multiplication tables etc. Checking for primes, generation of Armstrong numbers. Generation of the Fibonacci sequence, Finding the square root of a number, calculation of factorials, printing various patterns using for loop. The greatest common divisor of two integers, Raising a number to large power.

Unit-IV: Programs using Arrays: declaring and initializing arrays. Program to do simple operations with arrays. Strings – inputting and outputting strings. Using string functions such as strcat, strlen etc. Writing simple programs for strings without using string functions.Finding the maximum number in a set, Array order reversal, Finding maximum number from an array of numbers Removal of duplicates from an ordered array,

Unit-V: Selection/ Bubble/ Insertion sort, create a linked list, traverse a linked list, insert a node and delete a node form the list. Recursion and related examples such as Tower of Hanoi, computing factorial etc. Practice sessions and sessions for missed labs

Units to Lab Mapping:

Unit	**Labs**
I	1, 2, 3
II	4, 5
III	6, 7, 8
IV	9, 10, 11
V	12, 13, 14

Teaching Methodology:

This course is introduced to help students understand the discipline of programming. The programming language used to teach this course is C. Starting from the programming environment setup, the student will slowly be exposed to program designing and later to programming fundamentals. The entire course is broken down into six separate units, from fundamentals of programming to some complex programming structures like pointers. This theory course is well complemented by a laboratory course under the name Software Development Fundamentals Lab in the same semester that helps a student learn with hand-on experience.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
P-1	15 Marks	**Based on Lab Exercises: 1-6**

P-2		15 Marks	**Based on Lab Exercises: 7-13**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Learning Resources:

Study material of Software Development Fundamentals Lab (will be added time to time): Digital copy will be available on the JUET server.

Text Book:

1. Programming in ANSI C by E. Balguruswamy, Tata Mc-Graw Hill.
2. Programming With C, Schaum Series.

Reference Books/Material:

1. The 'C' programming language by Kernighan and Ritchie, Prentice Hall
2. Computer Programming in 'C' by V. Rajaraman, Prentice Hall
3. Programming and Problem Solving by M. Sprankle, Pearson Education
4. How to solve it by Computer by R.G. Dromey, Pearson Education

Web References:

1. http://www2.its.strath.ac.uk/courses/c/
   a. Notes on C programming by University of Strathclyde Computer Centre. This tutorial was awarded the NetGuide Gold Award during the 1990s.

2. http://www.princeton.edu/~achaney/tmve/wiki100k/docs/C_%28programming_language%29.html
   a. This site contains notes on C programming from Princeton University, USA. These are very useful for students who are learning C as their first programming Language.

3. http://www.stat.cmu.edu/~hseltman/Computer.html
   a. Online reference material on Computers and Programming from Carnegie Mellon University, Pittsburgh, USA

4. http://projecteuler.net/
   a. Collection of mathematical problems which make you use your programming skills

Title: Workshop Practices **Code: ME201**
L-T-P scheme: 0-0-3 **Credit: 1.5**

Prerequisite: Students must have the knowledge of fundamental principles of Physic and Chemistry upto class 12^{th} which helps them to understand the various process of Workshop Lab.

Objective:

1. To demonstrate students, the basic manufacturing processes of Workshop lab: Carpentry, Fitting, Welding, Machining and Casting Processes.
2. To develop effective skills in students to identify the manufacturing process with its applications
3. To be able to perform basic manufacturing processes safely.

Learning Outcomes:

Course Outcome	**Description**
CO1	Identify the various processes of manufacturing.
CO2	Capable to explain the use of various holding, measuring, marking and cutting tools of workshop
CO3	Prepare a useful job by performing the various processes in proper sequence safely
CO4	Apply Bernoulli's theorem to analyze the liquid metal velocity in casting process.
CO5	Develop the skills to join two metallic specimen using welding process
CO6	Work as a team on a project

Course Content:

Carpentry Shop

1. To study about various tools/equipments used in carpentry shop
2. To make Cross lap /T joint as per given specification
3. To make Cross lap /T joint as per given specification

Foundry Shop

1. To study about various tools used in foundry shop.
2. To prepare a green sand mould with the help of a given pattern.
3. To perform permeability test on moulding sand

Machine Shop

1. To study various machine tools such as lathe, milling, shaper, drilling, grinding, EDM drill and cutting tools used by them.
2. To perform turning, step turning and taper turning operations on lathe machine
3. To perform threading operation on the lathe machine

Fitting Shop

1. To study about various tools used in fitting shop.
2. To make a fitting job as per given drawing.

Welding Shop

1. To study various types of welding processes available in the workshop such as Electric arc welding, TIG and MIG welding, gas welding and spot resistance welding,
2. To prepare welding joint by using Electric arc welding/gas welding
3. To prepare welding joint by using Spot Resistance welding

Teaching Methodology:

This Lab course has been introduced to help a student to learn with hand-on experience on machines. The entire course is broken down into fourteen experiments. Experiments are performed different shop wise by taking the proper safety precautions. Workshop lab includes five shops namely: Carpentry, Foundry, Machining, Fitting and Welding. Basic principles of manufacturing processes are applied to prepare a job. Students learn here how to handle the real world problems by using technical skills. The way of experimentation here realizes the students that they are now moving on an Engineering path. This Lab course will enable a student to learn with hand-on experience.

Evaluation Scheme:

<table>
<tr><th colspan="2">Exams</th><th>Marks</th><th>Coverage</th></tr>
<tr><td colspan="2">P-1</td><td>15 Marks</td><td>Based on Lab Experiments: 1-7</td></tr>
<tr><td colspan="2">P-2</td><td>15 Marks</td><td>Based on Lab Experiments: 8-14</td></tr>
<tr><td rowspan="4">Day-to-Day Work</td><td>Viva</td><td>20 Marks</td><td rowspan="4">70 Marks</td></tr>
<tr><td>Demonstration</td><td>20 Marks</td></tr>
<tr><td>Lab Record</td><td>15 Marks</td></tr>
<tr><td>Attendance & Discipline</td><td>15 Marks</td></tr>
<tr><td colspan="2">Total</td><td colspan="2">100 Marks</td></tr>
</table>

Learning Resources:

Laboratory Manual available in Lab. Study material of Workshop Lab (will be added time to time): Digital copy will be available on the JUET server.

Text Books:

[1] "Workshop Technology Volume- I & II", B.S. Raghuvanshi, Dhanpat Rai & Co.

[2] "Workshop Technology Volume-I & II", Khanna Publisher.

Reference Books:

[1] "Workshop Technology Vol.- 1, 2, 3 & 4", Butterworth-Heinemann.

[2] "Material Science & Engineering", W. D. Callister, John Wiley

Web References:

[1] https://nptel.ac.in/courses/112/107/112107219/

[2] https://nptel.ac.in/courses/112/107/112107144/

IInd Semester

Title: Life Skills and Effective Communication **Code: HS104**
L-T-P scheme: 1**-1-0** **Credit: 2**

Prerequisites: None

Objective:

1. To employ positive behavior management techniques and to develop skills to manage their own behavior effectively
2. To develop one's personality by being aware of the self, connecting with others, reflecting on the abstract and the concrete.
3. To enhance the employability and maximize the potential of the students by introducing them to the principles that underlying personal and professional success, and help them acquire the skills needed to apply these principles in their lives and careers.

Learning Outcomes:

CO	Outline different life skills required in personal and professional life.
CO	Describe the application of different theoretical perspectives within the field of motivation and applying these motivation theories to everyday settings (e.g., business, social interactions, education)
CO	Develop the understanding of personality and shaping behavior through personality
CO	Identify the basic mechanics of perception by demonstrating these through presentations.
CO	Apply well-defined techniques to cope with emotions and stress and develop an awareness of the self.

CO	Understand the basics of leadership and Learning

Course Content:

Unit-1: Overview of Life Skills: Meaning and significance of life skills, Life skills identified by WHO: Self-awareness, Empathy, Critical thinking, Creative thinking, Decision making, problem solving, Effective communication, interpersonal relationship, coping with stress, coping with emotion.

Unit-2: Motivation: Morale and Morale Building, Need and Importance of motivation, Process and types of motivation, Theories of motivation, Essentials of Good Motivation system

Unit-3: Overview of Personality concept and types, Personality traits, Factors that help in shaping personality, Theories of personality, Measurement of personality

Unit-4: Perception: - Factors affecting perception, Perceptual mechanisms Perceptual errors and distortions, Behavioral applications of perceptions

Unit-5: Self Awareness, Coping with emotions: Identifying and managing emotions, harmful ways of dealing with emotions, Stress Management: Stress, reasons and effects, identifying stress, Managing Stress

Unit-6: Conflict Management –sources, process and resolution of conflict

Unit-7: Leadership: Need for Leadership, Models of leadership development, and Characteristics of a good leader.

Unit-8: Learning: Concepts and Theories, classical conditioning, operant conditioning, Biological influences, Cognitive influences, Social learning theory, Behavioral modification theory

Teaching Methodology:

Life skills are those competencies that provide the means for an individual to be resourceful and positive while taking on life's vicissitudes .This course will equip students with the social and interpersonal skills that enable them to cope with the demands of everyday life. There will be a particular focus on social-cognitive processes and how situational factors trigger various emotions and corresponding motives that can then drive behavior. The main objectives of this course is to build self-confidence, encourage critical thinking, foster independence and help students to communicate more effectively

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Based on Unit-1, Unit-2 & Unit-3
Test-2	25 Marks	Based on Unit-4 & Unit-5 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-6, Unit-7 & Unit-8 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Case studies, video lectures and lecture slides on Life Skills (will be added from time to time): Digital copy will be available on the JUET server.

Text Book:

1. "Effective Communication and Soft Skills"; Nitin Bhatnagar, Pearson Education India,1e, 2011
2. "Personality Development and Soft Skills"; Barun Mitra, Oxford Higher Education,2016
3. "Sizzling Soft Skills for Spectacular Success"; P. Ameer Ali, Notion Press, 2017
4. "Organizational Behavior"; Stephen P. Robbins, Timothy A. Judge, Neharika Vohra, Pearson Education India, 16e, 2016
5. "Managing Organisations"; Rachna Chaturvedi, Vikas Publications, 2013

Reference Books/Material:

1. "The Power of Your Subconscious Mind"; Joseph Murphy, General press, 2015
2. "The Life-Changing Magic of Tidying Up: The Japanese Art of De cluttering and Organizing"; Marie Kondō, 1e,Ten speed Press, 2011
3. "The Power of Habit: Why We Do What We Do in Life and Business"; Charles Duhigg, Random House, 2012

Course Title: Discrete Mathematics **Course Code: MA105**

L-T-P scheme: 3-1-0 **Credits: 4**

Objectives:

The aim of the course is to cover the basic principles sets relations functions partially ordered set, lattice, Boolean algebra and its applications. The main objective of the course is to develop in student, an intuitive understanding of graphs by emphasizing on the real world problems.

Course Outcomes:

At the end of the course, the student is able to:

CO1	Employ De Moivre's theorem in a number of applications to solve numerical problems.
CO2	Appreciate the definition and basics of graphs along with types and their examples.
CO3	Visualize the applications of graph theory to network flows. Understand the notion of planarity and coloring of a graph. Relate the graph theory to the real-world problems.
CO4	Understand the definition of a tree and learn its applications to fundamental circuits.
CO5	Solve real-life problems using finite-state and Turing machines
CO6	Learn about partially ordered sets, lattices and their types, Boolean algebra and Boolean functions, logic gates, switching circuits and their applications.

Course Contents:

Unit 1: Basics of set theory, Mathematical induction. Relations, Equivalence relation, partial- ordered relation algorithms and functions.

Unit 2: Big O notation, Proposition, Basic logical operators, Propositional functions and Quantifiers.

Unit 3: Graphs and related definitions, Eulerian and Hamiltonian graphs, Graph colorings. Trees, Algebraic expressions and Polish notation, shortest path.

Unit 4: Algebraic Systems. Lattice and Boolean Algebra.

Unit 5: Language, Finite State Automata and Machines. Grammars.

Methodology:

The course will be covered through lectures supported by tutorials. Apart from the discussions on the topics covered in the lectures assignments/ quizzes in the form of questions will also be given.

Evaluation plan:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1
Test-2	25 Marks	Based on Unit-2 & Unit-3 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-4 to Unit-5 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

References:

1. B. A, Davey & H. A. Priestley (2002). "Introduction to Lattices and Order" (2nd edition) Cambridge University, Press.
2. Edgar, G. Goodaire & Michael M. Parmenter (2018). "Discrete Mathematics with Graph Theory" (3rd edition). Pearson Education.
3. Rudolf Lidl & Günter Pilz (1998). "Applied Abstract Algebra" (2nd edition).Springer.
4. Kenneth H. Rosen (2012). "Discrete Mathematics and its Applications: With Combinatorics and Graph Theory" (7th edition), McGraw-Hill.
5. C. L. Liu (1985). "Elements of Discrete Mathematics" (2nd edition). McGraw- Hill.

Title of Course: Engineering Physics-II **Course Code: PH102**

L-T Scheme: 3-1-0 **Course Credits: 4**

Objective:

Broadly, the study of Physics improves one's ability to think logically about the problems of science and technology and obtain their solutions. The present course is aimed to offer a broad aspect of those areas of Physics which are specifically required as an essential background to all engineering students for their studies in higher semesters. At the end of the course, the students will have sufficient scientific understanding of basic vector calculus, electrostatics, magnetostatics, electromagnetic fields and waves, basic understanding of physics of semiconducting materials

Course Outcomes:

Course Outcome	**Description**
CO1	Learn to apply the basic concepts of vector calculus and understanding of various co-dinate systems and related properties, Demonstrate basic understanding of formulation and duction of electric field produced by static charge distributions
CO2	Evaluate the electrostatic field due to symmetric charge distributions, Understand the utility of formulation of electric potential and solve related problems using special techniques and boundary conditions

CO3	Acquired understanding of electrostatic fields inside matter, Explain the magnetic field due to moving charge distribution, evaluate the magnetic field due to current distribution in space,
CO4	appreciate the importance of Maxwell's equations and understand the electromagnetic wave propagation in free space Categorisation of materials on the basis of band structure
CO5	Developed understanding of quantum mechanical origin of band formation in solids, describing the energy state of electrons in crystalline materials, comprehend basic carrier properties

Course Content:

Unit I *(Electrostatics)*

Review of vector calculus, Cartesian, spherical polar and cylindrical co-ordinate systems, concept of gradient, divergence and curl, Coulomb's law, Gauss law and its applications, Boundary condition on electrostatic field, electric potential, Laplace equation, Poisson equation and related boundary value problems, capacitance, electrostatic fields in matter [**10**]

Unit II (*Magnetostatics*)

Lorentz force, cyclotron formula, line, surface and volume currents, , Biot-Savart law and its applications, Ampere's law and its applications, equation of continuity, Faraday's law of electromagnetic induction, boundary conditions on magnetic field, Magnetic field in matter [**08]**

Unit III (Electromagnetic field)

Maxwell's equations in free space and matter, Maxwell correction to Ampere's law, Electromagnetic waves in free space and matter, Transverse nature of em waves and Polarization, Propagation of electromagnetic field in free space and Poynting vector, Poynting theorem , Normal incidence of em waves [**10]**

Unit IV *(Elements of Solid State Physics)*

Basic ideas of bonding in solids, Crystal structure, X-ray diffraction, Band theory of solids, Distinction between metals, semiconductors and insulators **[04]**

Unit V (*Physics of Semiconductors)*

Band theory of solids, Kronig Penney model, effective mass, Direct and indirect bandgap semiconductors, optical and thermal properties, Fermi-Dirac Distribution in semi-conductors, Equilibrium carrier concentrations in intrinsic and extrinsic semiconductors, Fermi energy variation with temperature and impurity concentration, Hall Effect in semiconductors, P-N junction characteristics **[10]**

Text/ Reference Books:

1. D.J. Griffiths, *Introduction to electrodynamics*, Prentice Hall of India Ltd.
2. B.G. Streetman, S. Banerjee, Solid State Electronic Devices
3. Semiconductor Physics and Devices, Donald A. Neamen
4. Boylstad and Nashelsky, *Electronic Devices and Circuits*, PHI, 6e, 2001.
5. J. Reitz, F. Milford and R. Christy, *Foundation of Electromagnetic Theory*, Narosa Publishing.
6. J. Millman and C.C. Halkias, Electronic Devices and Circuits, Millman, McGra-Hill

Title: Electrical Science **Code: EC101**

L-T-P Scheme: 3-1-0 **Credit: 4**

Prerequisite: Students must have studied the core concepts of "*Physics-1*".

Course Objectives:

1. This course is designed for developing the understanding about basics of electrical and electronics concepts.
2. In this course students will have an enough idea about the working of systems and enable them to analyze a circuit.

Learning Outcomes:

1. The students shall acquire the generic skills to study & analyze the electrical and electronic systems.
2. This course will enable them to think and design various applications of the electrical and electronics at basic level.

The student will be able to:

Course Outcome	Description
CO1	Understand the basic electrical and electronics component and their importance determine the current, voltage and power.
CO2	Apply networks laws and theorems to solve electric circuits and may understand circuit reduction techniques with their advantages.
CO3	Understand charging discharging Steady state and transient
CO4	Demonstrate the use of semiconductor diodes in various applications.
CO5	Discuss and explain the working of transistors Amplifiers, their configurations and applications.
CO6	Analysis concept and two port networks simplification technique.

Course Content:

Unit I: Basic Electrical Circuit: Electromotive Force (EMF), Terminal Voltage; Resistance (*R*), Inductance (*L*) and Capacitance (*C*) from (i) Circuit, (ii) Energy, and (iii) Geometrical Points of View; Voltage Divider, Current Divider; Star-Delta Transformation; Voltage Source and Current Source, Source Transformation, Combination of Sources; Controlled (Dependent) Sources.

Unit 2: Methods of Analysis: Kichhoff's Circuit Laws; Loop-Current Analysis, Mesh Analysis; Node-Voltage Analysis; Choices of Method of Analysis.

Unit 3: Network Theorems (DC Circuits): Superposition Theorem; Thevenin's Theorem; Norton's Theorem; Maximum Power Transfer Theorem.

Unit 4: DC Transients: Simple *RL* Circuit, Time Constant, Decay and Growth of Current; Simple *RC* Circuit, Discharging of a Capacitor, Charging of a Capacitor.

Unit 5: Two-Port Networks: Impedance, Admittance, Hybrid, Transmission Parameters; Equivalent Networks.

Unit 6: Diodes and its Applications: Unidirectional property, *PN*-junction with no bias, with forward bias and with reverse bias, *V-I* characteristics, Comparison of Si and Ge diodes,Temperature effects, Diode resistance (static and dynamic), Diode equation, Ideal diode, Circuit model of a diode. Half-wave and full-wave (centre tap and bridge) rectifiers, PIV rating of diode, Performance of half-wave and full-wave rectifiers, Shunt capacitor filter. Clippers: Series and Parallel, Limiters, Clampers. Zener diode, Analysis of Zener voltage regulator. LED, varactor diode .

Unit 7: Transistor: BJT Structure, Working of a transistor, Transistor current equation, Collector reverse saturation current, DC alpha of a transistor. The three configurations, CB and CE input and output characteristics.

Teaching Methodology:

Lectures would be interactive and it would cover the core concepts that are explained in the text and reference materials with adequate examples.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15	Based on Unit-1 & Unit-2
Test-2	25	Based on Unit-3, Unit-4 & Unit-5 and around 30% from coverage of Test-1
Test-3	35	Based on Unit-6 to Unit-7 and around 30% from coverage of Test-2
Assignment	10	Based on Unit-1, Unit-2 & Unit-3
Tutorials	5	Based on Unit-4 & Unit-5
Quiz	5	Based on Unit-6 & Unit-7
Attendance	5	Based on attendance in the theory classes
Total	**100**	

Learning Resources:

Tutorials sheets, lecture slides and handwritten notes on Electrical circuit, Electrical Science and Basic Electronics (will be added from time to time): Digital copy will be available on the JUET server.

Text-Books:

1. D.C. Kulshreshtha, "Basic Electrical Engineering", McGraw Hill Education, 2009.
2. W.H. Hayt, J. E. Kemerlay & S.M. Durbin, "Engineering Circuit Analysis (Sixth Edition)", McGraw Hill, 2006.
3. R.C. Dorf & J.A. Svoboda, "Introduction to Electric Circuits", John Wiley, 2004.
4. D.S. Chauhan & D.C. Kulshreshtha, 'Electronics Engineering', New Age, 2e, 2009.
5. D.C. Kulshreshtha, 'Electronic Devices and Circuits', New Age, 2e, 2006.

References:

1. Van Valkenburg, "Network Analysis", Prentice-Hall India Ltd., 2001.
2. Abhijit Chakrabarti, Sudipta Nath, Chandan Kumar Chanda, "Basic Electrical Engineering", Tata McGraw Hill Publishing Co, 2008.
3. Vincent Del Toro, "Principles of Electrical Engineering", Prentice Hall of India.
4. Kumar and Jain, 'Electronic Devices and Circuits', PHI, 2007.
5. Boylstad and Nashelsky, 'Electronic Devices and Circuits', PHI, 6e, 2001.

Web References:

1. https://www.electrical4u.com/electrical-engineering-objective-questions-mcq/

2. https://www.pdfdrive.com/basic-electric-circuit-analysis-books.html
3. https://lecturenotes.in/subject/842

Journals References:

1. Circuits, Systems, and Signal Processing (CSSP), Springer
2. Journal of Electrical & Electronic Systems
3. International Journal of Circuit Theory and Applications, Wiley

Title of Course: Object Oriented Programming **Course Code: CS102**
L-T-P Scheme: 3-1-0 **Course Credit: 4**

Prerequisites:
Students must have already registered for the course, "Software Development Fundamentals"

Objectives:
To strengthen their problem solving ability by applying the characteristics of an object-oriented approach and to introduce object oriented concepts in C++.

Learning Outcomes

Course Outcome	Description
CO1	List various principles of Object-Oriented Programming (OOP).
CO2	Describe the real world problems using object-oriented programming concepts.
CO3	Develop the programs using the fundamental concepts of OOP.
CO4	Identify and use various techniques used in OOP.
CO5	Apply techniques used in OOP to solve the software design problems on a given software project.

CO6	Demonstrate the learning on the course to solve the real life programming problems.

Course Content

Unit-1: Review of Structured programming in C, Structured versus Object-Oriented Programming, Principles of Object-Oriented Programming, Beginning with C++, Control Structures, Functions in C++, Reference Variables, Default Parameters, Function Overloading, Inline Function, Const Variables.

Unit-2: Classes, Member Functions, Objects, Static Data Members, Static Member Functions, Friend Functions, Pointer to Members, Local classes, Constructors and Destructors of objects in C++,

Unit-3: Operator overloading and Type Conversions, Inheritance and its form, Multiple Inheritance in C++, Function Overriding, Virtual Inheritance, Virtual Base Class .

Unit-4: Pointers, Early binding, late binding, Type of polymorphism, Virtual Functions, Abstract Class, Virtual Destructor

Unit-5: Managing Console I/O Operations, File handling and Exception handling.

Unit-6: Templates, Function templates, Class templates, introduction to Standard Template Library (STL), Sequence, Containers, Iterators

Teaching Methodology

The course will use the mixed technique of interactive lectures, tutorials, guided case studies, literature survey, regular assignments and project work. Teaching in this course is designed to engage the students in active and experiential learning by taking a problem solving and design-oriented approach with special emphasis on real world applications.

In the lectures the fundamental theoretical concepts will be introduced and demonstrated through examples and case studies. Discussion in lecture will be done using design problems which will be implemented in laboratory individually in C++.

Evaluation Scheme

Evaluations	**Marks**	**Remarks**
T1	15 Marks (1 Hour)	
T2	25 Marks (1.5 Hours)	
T3	35 Marks(2 Hours)	
Assignments	10 Marks	2 or 3 Assignments to given
Quiz	5 Marks	2 or 3 quizzes
Tutorials	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Text books

Text book1: Robert Lafore, Object oriented programming in C++, Waite Group.
Text book2: E Balagurusamy, "Object-Oriented Programming with C++"

References

1. Deitel and Deitel,"C++ How to program", - Pearson Education.
2. Stroustrap B., the C++ Programming Language, Addison Wesley.
3. Lippman F. B., C++ Primer, Addison Wesley.
4. Prata S., C++ Primer Plus, Waite Group.
5. Parimala N., Object Orientation through C++, Macmillan India Ltd. 1999.
6. Pohl I., Object oriented Programming Using C++, Addison Wesley.
7. Grady Booch, James Rambaugh, Ivar Jacobson,"Unified Modelling Language user's guide" , Addison Wesley Limited

Title of Course: Engineering Physics Lab-II **Course Code: PH202**
L-T-P Scheme: 0-0-2 **Course Credit: 1**

Learning Outcomes

Course Outcome	**Description**
CO1	Demonstrate ability to collect experimental data and understanding the working procedures within the precautionary limits
CO2	Acquired the ability to analyze the experimental data and related errors in a reflective, iterative and responsive way
CO3	Developed understanding of the basic concepts related to Modern Physics, Basic Solid State Physics, Optics,
CO4	Acquired a first hand and independent experience of verifying the working principle of solar cell

CO5	Appreciate the importance of the laboratory work culture and ethics that is intended to impart features like regularity, continuity of self evaluation and honesty of reporting the data

Experiments List

1. To determine the magnetic susceptibility of a paramagnetic, $FeCl_3$ solution by Quinck's tube method.
2. To determine dispersive power of a prism using spectrometer.
3. To study the magnetostriction in metallic rod using Michelson-Interferometer.
4. To determine the Planck's constant using Photo electric effect.
5. To study the Hall effect in P type semi conductor and to determine
   (i) Hall voltage and Hall coefficient
   (ii) Number of charge carriers per unit volume
   (iii) Hall angle and mobility
6. To study the variation of resistivity of a semiconductor with temperature and to determine the band gap using Four-Probe method.
7. To study the presence of discrete energy levels in an atom by Franck Hertz experiment.
8. Using solar cell Trainer (a) study voltage and current of a solar cell (b) Voltage and current in series and parallel combinations (c) Draw power curve to find maximum power point (MPP) and to obtain efficiency of a solar cell.

Title: Electrical Science Lab **Code: EC203**
L-T-P Scheme: 0-0-2 **Credit: 1**

Prerequisite: Student must have already registered for the course, "*Physics Lab-I*"

Objective:

1. The main aim of the lab is to familiarize with different types of electrical and electronic circuits
2. Identify their applications to the different electrical and electronic systems.

Learning Outcomes:

1. Completion of lab students will be able to understand the different techniques to simplify circuit

2. Two port networks and basic principles of different electronic devices and their characteristics.

Course Outcome	Description
CO1	Simplify complex network using Thevenin theorem and verify it.State Superposition Theorem and verify.Perform and verify Maximum Power Transfer Theorem.
CO2	To determine the Z parameters of the given two port network. Calculate the Y parameters for the given two port network.
CO3	V-I characteristic of p-n junction diode
CO4	Design Clipper and Clamper Circuit.
CO5	Rectifier circuits
CO6	Transistor and their v-I characteristics

Course Content:

1. Simplify complex network using Thevenin theorem and verify it.
2. State Superposition Theorem and verify.
3. Perform and verify Maximum Power Transfer Theorem.
4. To determine the Z parameters of the given two port network.
5. Calculate the Y parameters for the given two port network.
6. Perform Clipper Circuit.
7. Design Clamper Circuit.
8. Half wave rectifier with and without filter circuit.
9. Full wave rectifier with and without filter circuit.
10. Transistor as an Amplifier.
11. Common Emitter *v-i* characteristic of n-p-n transistor.
12. Common base *v-i* characteristic of n-p-n transistor.

Unit I: Basic Electrical Circuit

Voltage Divider, Current Divider; Kichhoff's Circuit Laws; Loop-Current Analysis, Mesh Analysis; Node-Voltage Analysis; Choices of Method of Analysis. Source Transformation, Combination of Sources; series and parallel combination of resistors.

Unit 2: Network Theorems (DC Circuits)

Superposition Theorem; Thevenin's Theorem; Norton's Theorem; Maximum Power Transfer Theorem.

Unit 3: Two-Port Networks

Impedance, Admittance, Hybrid, Transmission Parameters; Equivalent Networks.

UNIT 4: Diodes and its Applications

Unidirectional property, *PN*-junction with no bias, with forward bias and with reverse bias, *V-I* characteristics, Diode resistance (static and dynamic), Diode equation, Ideal diode, Circuit model of a diode. Half-wave and full-wave (centre tap and bridge) rectifiers, PIV rating of diode, Performance of half-wave and full-wave rectifiers, Shunt capacitor filter.

Clippers: Series and Parallel, Limiters, Clampers. Zener diode, Analysis of Zener voltage regulator. LED, varactor diode .

UNIT 5: Transistor

BJT as an amplifier, CB and CE input and output characteristics.

Teaching Methodology:

In each experiment the practical is designed and analyzed on bread board with the help of physical devices by each student and further checked and validated by faculty and lab staff.

Evaluation Scheme:

Exams	**Marks**		**Coverage**
P-1	**15 Marks**		**Based on Lab Exercises: 1-6**
P-2	**15 Marks**		**Based on Lab Exercises: 6-12**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Learning Resources:

Tutorials sheets, lecture slides and handwritten notes on Electrical circuit, Electrical Science and Basic Electronics (will be added from time to time): Digital copy will be available on the JUET server.

Text-Books:

1. D.C. Kulshreshtha, “Basic Electrical Engineering”, McGraw Hill Education, 2009.
2. W.H. Hayt, J. E. Kemerlay & S.M. Durbin, “Engineering Circuit Analysis (Sixth Edition)”, McGraw Hill, 2006.
3. R.C. Dorf & J.A. Svoboda, “Introduction to Electric Circuits”, John Wiley, 2004.
4. D.S. Chauhan & D.C. Kulshreshtha, ‘Electronics Engineering’, New Age, 2e, 2009.
5. D.C. Kulshreshtha, ‘Electronic Devices and Circuits’, New Age, 2e, 2006.

References:

1. Van Valkenburg, “Network Analysis”, Prentice-Hall India Ltd., 2001.
2. Abhijit Chakrabarti, Sudipta Nath, Chandan Kumar Chanda, “Basic Electrical Engineering”, Tata McGraw Hill Publishing Co, 2008.
3. Vincent Del Toro, “Principles of Electrical Engineering”, Prentice Hall of India.
4. Kumar and Jain, ‘Electronic Devices and Circuits’, PHI, 2007.

5. Boylstad and Nashelsky, 'Electronic Devices and Circuits', PHI, 6e, 2001.

Web References:

1. https://www.electrical4u.com/electrical-engineering-objective-questions-mcq/
2. https://www.pdfdrive.com/basic-electric-circuit-analysis-books.html
3. https://lecturenotes.in/subject/842

Journals References:

1. Circuits, Systems, and Signal Processing (CSSP), Springer
2. Journal of Electrical & Electronic Systems
3. International Journal of Circuit Theory and Applications, Wiley

Title of Course: Object Oriented Programming Lab **Course Code: CS202**
L-T-P Scheme: 0-0-2 **Course Credit: 1**

Pre-requisites

Students must have already registered for the course, "Software Development Fundamentals Lab".

Objectives

To strengthen their problem solving ability by applying the characteristics of an object-oriented approach and to introduce object oriented concepts in C++.

Learning Outcomes

CO1	Define basic concepts of Object-Oriented Programming (OOP).

CO2	Illustrate the key features available in OOP using C++.
CO3	Apply the concepts of OOP to solve different common problems.
CO4	Utilize the knowledge of OOP in solving programming problems.
CO5	Analyze the various concepts of OOP for their suitability on a given problem.
CO6	Design the systems, from concept to executable artefact, using object oriented techniques.

Course Content

Unit-1: Structured versus Object-Oriented Programming, Principles of Object-Oriented Programming, Beginning with C++, Control Structures, Functions in C++, Reference Variables, Default Parameters, Function Overloading, Inline Function, Const Variables.

Unit-2: Classes, Member Functions, Objects, Static Data Members, Static Member Functions, Friend Functions, Pointer to Members, Local classes, Constructors and Destructors of objects in C++,

Unit-3: Operator overloading and Type Conversions, Inheritance and its form, Multiple Inheritance in C++, Function Overriding, Virtual Inheritance, Virtual Base Class .

Unit-4: Pointers, Early binding, late binding, Type of polymorphism, Virtual Functions, Abstract Class, Virtual Destructor

Unit-5: Managing Console I/O Operations, File handling and Exception handling.

Unit-6: Templates, Function templates, Class templates, introduction to Standard Template Library (STL), Sequence, Containers, Iterators

Laboratory work and project

The students shall be given regular lab assignments, which will allow them to practically apply the concepts studied in the lecture Session. The lab assignments will be designed with focus on applying the concepts learnt in object-oriented programming, Data structures in an integrated manner.

Evaluation Scheme

Evaluations		**Marks**	**Remarks**
P-1		15 Marks	
P-2		15 Marks	
Continuous Evaluations	Viva	20 Marks	
	Demonstration	20 Marks	
	Lab Record	15 Marks	

	Discipline and Punctuality and Attendance	15 Marks	
Total		**100 Marks**	

Text book

Text Book1: Robert Lafore, Object oriented programming in C++, Waite Group
Text Book2: E Balagurusamy, "Object-Oriented Programming with C++"

References

1. Stroustrap B., the C++ Programming Language, Addison Wesley.
2. Lippman F. B., C++ Primer, Addison Wesley.
3. Prata S., C++ Primer Plus, Waite Group.
4. Parimala N., Object Orientation through C++, Macmillan India Ltd. 1999.
5. Pohl I., Object oriented Programming Using C++, Addison Wesley.
6. Grady Booch, James Rambaugh, Ivar Jacobson,"Unified Modelling Language user's guide" , Addison Wesley Limited

Title: Engineering Drawing & Design Lab **Code: ME203**
L-T-P scheme: 0-0-3 **Credits: 1.5**

OBJECTIVE

[1] Enables students to learn the concepts of graphic communication, their role in sanitary construction.
[2] Make familiar with different drawing equipment, technical standards and procedures for construction of geometric figures.
[3] Equipped with the skill that enables them to convert pictorial to orthogonal representations.

Learning Outcomes:

Course Content:

Unit-1: Study and construction of lines, lettering, dimensioning, plane scales, diagonal scales, construction of different methods used for the construction of conic curves.
Unit-2: Study and construction of geometrical construction, cycloidal curves, involutes and helix etc.
Unit-3: Orthogonal projection of point in all possible positions, Study and construction of projection of line and its applications (inclined to both planes), and projection of planes (inclined to both planes).
Unit-4: Study and construction of projection of solids (right circular cone, prism, pyramid and cylinders), and true shape of sections,

Course Outcome	**Description**
CO1	**Outline** the objectives of scale and develop the imagination and mental visualization capabilities for correlating the geometrical details of objects.
CO2	To **develop** the constructional ability for a different curve.
CO3	To **Describe** BIS rules for orthogonal projection and understand the fundamental concept of orthogonal projection for point, line, plane and solids.
CO4	Understand and **apply** orthogonal projection for solids, section and intersection of solid objects/structures
CO5	To **apply** the skill of development of surfaces of three dimensional objects for evaluation of black size of the components.
CO6	**Demonstrate** computer aided drafting tools and techniques using CAD software's

Unit-5: Study and construction of oblique projection and development of surface, isometric view using orthogonal projection on isometric scales.
Unit-6: Introduction to basic and editing command of CAD software, 2-D drafting, surface modeling, and 3-D geometrical model.

Teaching Methodology:
This course is introduced to build the imagination and established the correlation between the real object and engineering drawing and CAD developed by the design engineers and the requirement of the production engineers of the different units.
Evaluation Scheme:

Exams		Marks	Coverage
P-1		15 Marks	**Based on Lab Exercises: 1-7**
P-2		15 Marks	**Based on Lab Exercises: 8-14**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	

	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Learning Resources:
The study material of engineering drawing & design lab (will be added time to time): Digital copy will be available on the JUET server.

Text Book:

1. Bhatt, N.D., Engineering Drawing,

Reference Books:

1. Gill, PS, A Text Book of Engineering Drawing (Geometrical Drawing)
2. Dhananjay A J, Engineering Drawing with an introduction to Auto CAD, Mc Graw Hill

3rd Semester:

Title: Techniques for Decision Making **Code**: **HS103**
L-T-P scheme: 2-1-0 **Credit: 3**

Prerequisite: None

Objectives:

1. To use basic techniques of inferential data analysis, quality control, and regression modeling;
2. To analyze a set of data, to reach a conclusion based on these analyses, and to make and defend a recommended course of action;
3. To be well-equipped to take courses in Marketing, Investments, Accounting, Finance, and Operations Management that require proficiency in statistical methods.

Learning Outcomes:

Course Outcome	Description
CO1	Outline various concepts of techniques for decision making with respect to the needs of modern business management.
CO2	Describe the real world problems using basic techniques of descriptive and inferential data analysis and business forecasting.
CO3	Identify and use various index numbers used in business decision making.
CO4	Apply decision making techniques to reach a conclusion based on the data analysis, and to make and defend a recommended course of action.
CO5	Deployment and proficiency in statistical methods.
CO6	Develop the understanding to analyze a set of data using correlation analysis and regression analysis.

Course Content:

Unit-1: Collection of data and Presentation of data: Classification of data, Secondary data, Primary data, Designing of questionnaire, Unstructured and structured questionnaire, Tabulation of data, Charting of data.

Unit-2: Business Forecasting: Introduction, steps in forecasting, good forecasting, Time series forecasting, secular trend, seasonal variations, cyclical variations.

Unit-3: Index numbers: Uses, classification, problems, Methods of constructing index numbers, unweighted index numbers, Consumer Price index numbers.

Unit-4: Statistical Decision making : Decision making under certainity, Risk , uncertainty and conflict, Zero sum game, Prisoner's dilemma , Payoff Table, Maximin and minimax strategy.

Unit-5: Correlation Analysis and Regression analysis: Significance of the study of correlation, Correlation and causation, Karl Pearson's coefficient of correlation, Rank correlation, Method of least squares, Difference between correlation and regression, Regression lines and regression equation, Regression equation of Y on X and regression equation of X on Y.

Teaching Methodology:

The course "Techniques for Decision Making" is introduced to explain the basic concepts in statistics that have wide applicability in business decision making. As such, the focus will be more practical than theoretical. Because statistical analysis informs the judgment of the ultimate decision-maker—rather than replaces it—we will cover some key conceptual underpinnings of statistical analysis to insure that the students understand its proper usage. Statistics is about improved decision-making, which can be achieved through a thorough understanding of the data. We want to leave our pre-conceived notions at the door, and let the data tell us what is going on in a situation. The analytical techniques should provide valuable information to decision-makers.

As such, it plays an important role in management decision processes. The course will be taught with the aid of lectures, tutorials, handouts, case studies, and problem-based learning.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1 & Unit-2
Test-2	25 Marks	Based on Unit-3 & Unit-4 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-5 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	100 Marks	

Learning Resources:

Lectures, tutorials and e-books on Techniques for Decision Making (will be added from time to time): Digital copy will be available on the JUET server.

Text Book:

1. "Business Statistics"; S.P. Gupta & M.P. Gupta, S. Chand Publishing, New Delhi, 2013.

Reference Books/Material:

1. "Statistics for Business & Economics"; Anderson, Thomson Learning, Bombay.
2. "Quantitative Methods in Business"; Anderson, Thomson Learning, Bombay.
3. "Business Statistics"; R.S. Bhardwaj, Excel Books.
4. "Statistics for Management"; Levin & Rubin, Prentice Hall of India, New Delhi.
5. "Two Person Game Theory"; A. Rapport & Anne Arbric, The University of Michigan Press, 1966.

Title of Course: Data Structures **Course Code: CS103**
L-T-P Scheme: 3-1-0 **Credits: 4**

Scope and Objectives:

This course develop problem solving ability using programming, develop ability to express solutions to problems clearly and precisely, develop ability to design and analyze algorithms, introduce with fundamental data structures, develop ability to design and evaluate abstract data types and data structures.

Learning Outcome:

The students shall acquire the generic skills to design and implement data structures and related algorithms for a broad-based set of computing problems**.**

18B11CI311: Data Structures	
Course Outcome	**Description**
CO1	List various types of data structures with respect to their requirements in different fields.
CO2	Describe the various methods to evaluate the algorithms.
CO3	Develop algorithms based on linear data structures
CO4	Identify the suitability of the data structures as per the requirements.
CO5	Apply data structures to solve the software design problems.
CO6	Demonstrate the learning on the course to solve the real life programming problems.

This course is intended to provide a thorough introduction to the use of data structures in programming. This course will cover the necessary mathematical background, but will assume the required programming experience.

Course Contents:

UNIT 1: Introduction to Data Structures, Algorithm and Complexity

Data structure overview, need of data structure and how to select relevant data structure for given problem, basic C data types and ADT.

Algorithm overview and its properties, problem analysis and construction of algorithm, difference between algorithm, program and software, algorithm analysis and complexity, asymptotic notations to represent the time complexity, Software Development Life Cycle (SDLC) phase

UNIT 2: Array

Overview, memory representation of 1D and 2D array, sparse matrix, operation supported by an array

Part 1: Searching

Linear search with illustration, analysis of linear search, binary search (iterative) and its analysis, binary search (recursive) and its analysis using recurrence relation, recurrence relation

Part 2: Sorting

Types of sorting algorithms, bubble sort, selection sort, insertion sort, quick sort, merge sort

UNIT 3: Linked List

Overview, types of linked list, linear linked list – overview, traversing, insertion, deletion, searching and reverse, doubly linked list – overview, traversing, insertion, deletion, circular linked list – overview, header linked list, applications of linked list

UNIT 4: Stack

Overview, stack implementation using stack and linked list, basic operations on stack using array and linked list – push, pop, dispose applications of stack – evaluation of mathematical expression, conversion of expression from one form to another (Polish Notation), Tower of Hanoi problem

UNIT 5: Queue

Overview, basic operations on queue – enqueue, dequeue, implementation of queue using array and linked list, types of queue - linear queue, circular queue, deque, priority queue, applications

UNIT 6: Tree

Tree definition and its terminology, representation of graph using array and linked list, tree traversals – preorder, inorder and postorder, binary search tree (BST) with insertion, deletion and searching operations, extended binary tree and its application in Huffman tree, threaded binary tree

UNIT 7: Graph

Introduction to graph, types of graph, traversal algorithms in graph – breadth first search, depth first search, spanning tree, minimum cost spanning tree - Kruskal's, Prim's.

Evaluation Scheme:

Component & Nature	Duration	Marks / Weightage
T1	1 hr	15
T2	1&1/2 hrs	25
T3	2hrs	35
Tutorials		05
Attendance		05
Quiz		05
Assignments		10
Total		**100**

Text Book::

T1: Sartaj Sahni, "Fundamentals of Data Structures", Tata Mc Graw Hill, New York
T2: Seymour Lipschutz., "Data Structures with C", Schaum's Outline Series
T3: Narasimha Karumanchi, "Data Structures and Algorithms" Made Easy

Reference Books:

R1: Corman et al: Introduction to Computer Algorithms
R2: Langsam, Augestein, Tenenbaum: Data Structures using C and C++
R3: Weiss: Data Structures and Algorithm Analysis in C/C++
R4: Samir K. Bandyopadhyay," Data Structures using C"
R5: Hopcraft, Ullman: Data Structures and Algorithms

Title of Course: Digital Systems and Microprocessors **Course Code: EC106**
L-T-P Scheme: 3-1-0 **Credits:4**

Course Objective:

Digital Systems and Microprocessor Course is the Second year's course which is totally based on study and designing Digital components, digital circuits using basic components, types of signals on which these devices works and at last the study of the Microprocessor basics in a single

course. This course aims to introduce students with a fundamental understanding of digital electronics and its application, Produce digital circuit, how signals are formed and further applications of microprocessor with all conditions.These undergraduate students will be equipped to play valuable roles in the Information Technology,Electronics and Communication industries.

Learning Outcomes:

Digital Systems and Microprocessor	
Course Outcome	**Description**
CO1	Outline various number systems of Digital Electronics with respect to the requirements of the computer systems used in technical industries fulfilling the user requirement.
CO2	Solving various problems based on the number systems, complements techniques, compute simple arithmetic operations addition, subtraction, multiplication & division including ability to prove implication problems using truth table method, Boolean method etc. considering the real world examples.
CO3	Design Karnaugh map and Quine McCluskey method to get simplified form of a Boolean function.
CO4	Design combinational and sequential digital functions.
CO5	Understanding the various types of signals used for the various explained devices and getting knowledge of trans-receiving the signals using explained devices.
CO6	Introduction of Microprocessor with its interfaces and basic coding understanding utilized in it. Understand the features and architecture of 16 bit Microprocessor.
CO7	Understand the data types and addressing modes of 8086 Microprocessor. Demonstrate deployment and basic maintenance skills.

Teaching Methodology:

Lectures would be interactive and it would cover the core concepts that are explained in the text and reference materials with adequate examples. Tutorials will have conceptual and numerical questions that would aid in strengthening the Digital electronics, signals and Microprocessors principles.Keeping in view the student's background, starting from number system to Basic pulse circuits design, the student will cover the study of basic signal types and application of microprocessors. In this course a student will learn about various digital components and designing digital circuits and moreover he will study about the various sequential and combinational circuits using basic gates and K-Map designing using the same gates. After this he will be taught combinational and sequential circuits which will make him proficient in designing any digital circuit. After this the basic knowledge of types of signals will be taught which will make them to learn how to implement these digital circuits over different types of signals and at last they will be taught about the Microprocessor basics which will guide them how Microprocessor world is more emphasizing on basics of Digital Electronics. And at the end of the course, successful students should have knowledge of and ability to apply the Mathematics and scientific concepts required by Digital Electronic engineers, basic level of knowledge of and

ability to apply the concepts, principles and theories of Computing and IT, as likely to be required by a Digital Electronic engineer, detailed knowledge of and ability to apply the essential facts, concepts, principles and theories needed by Digital Electronic engineers.

Course Outline:

Unit I:
Conversion of bases, Representation of negative numbers, 9's and 1's complement, 10's and 2's complement, Binary arithmetic, BCD code, Excess-3 code, Gray Code and Alphanumeric code. Logic gates and Boolean algebra, Standard and canonical representation and minimization of Boolean expressions using Karnaugh Map and Quine – McClausky methods.

Unit II:
Half & full adder and subtractor, Parallel adder, BCD adders, Lookahead carry generator. Decoders, Encoders, Multiplexers and De-multiplexers, Code convertor, Comparator, Parity generator and Checker. Binary multiplier.

Unit III:
Flip Flops: SR, JK, Master slave JK, T and D. Shift Registers and their Applications. Synchronous and Asynchronous counters, ROM, PROM, EPROM, EEPROM.

Unit IV:
Basics of Signals and Systems, Elements of a communication system, Continuous-time and discrete-time signals, signal energy and power, Periodic signals, even-odd signals, Exponential and Sinusoidal Signals.

Unit V:
Evolution of Microprocessor, Cache Memory, 8085 Architecture and its pin descriptions.

Evaluation Scheme:
Learning Resources:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Based on Unit-1, Unit-2 (30%)
Test-2	25 Marks	Based on Unit-2 (70%), Unit-3 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-4 & Unit-5 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Tutorials and lecture slides on Web Development (will be added from time to time): Digital copy will be available on the JUET server.

Text Books :

1. Morris Mano, Digital Logic and Computer Design, PHI
2. Taub and Schilling, Digital Integrated Electronics, McGraw Hill, Int. Ed.
3. Signal and Systems, 2nd Edition, PHI Publications, India 1997 by Alan V. Oppenheim, Alan S. Willsky, S. Hamid Nawab.

4. Fundamentals of Microprocessors and Microcontrollers, 7th edition, Dhanpat Rai Publication, India, 2010 by B. Ram.
5. Introduction to Microprocessors, Wiley Eastern (Latest Edition) R.S. Gaonkar.

Web References:

1. https://nptel.ac.in/courses/117106086/
2. http://web.iitd.ac.in/~shouri/eel201/lectures.php
3. https://www.geeksforgeeks.org/digital-electronics-logic-design-tutorials
4. https://www.electrical4u.com/digital-electronics

Journals References:

1. IEEE Transactions on Circuits and Systems
2. International Journal of Electronics by Taylor and Francis
3. AEÜ - International Journal of Electronics and Communications by Elsevier

Title of Course: Database Systems **Course Code: CS104**
L-T-P Scheme: 3-1-0 **Course Credits: 4**

Objectives: To develop the ability to design, implement and manipulate databases as well as to build Database management systems

Learning Outcome:

1. Ability to build normalized data bases.
2. Ability to design systems by using ER Modeling.
3. Ability to develop skills of writing applications by using SQL.
4. Ability to understand query optimization techniques.
5. Understanding of transaction processing.
6. Ability to handle recovery and concurrency issues

Course Outcome	Description
CO1	Introduction various types of database systems with respect to their features and charterstics and requirements in different fields.
CO2	Describe the various data definition, manipulation and various modifiers queries for database design.
CO3	Develop algorithms based on linear data structures
CO4	Develop the database using relational database query, Identify the suitable of the data structures as per the requirements.
CO5	Develop the normalized database with features of transaction, concurrency and recovery control
CO6	Demonstrate the learning on the course to deployed the database systems basis of the real life database problems.

Course Contents:

Introduction to Databases, Database Environment, Relational Model, Relational Algebra, SQL: Data Manipulation, Data Definition, And Commercial RDMS: MS-Access/MySQL, PL/SQL, ER Modeling: Entity type, Attributes, Relation types, Notations, Extended ER Features, Normalisation and building normalized databases & Data Dependencies, Case Study, Database Connectivity: Python MySQL Connectivity, Transactions, Concurrency, Recovery & Security, Query Processing & Optimization.

Text Book

1. “Database system concepts”, Henry F Korth, Abraham Silberschatz, S. Sudurshan, McGraw-Hill, 4th Edition.

References

1. “An Introduction to Database Systems” Bipin. C. Desai. Revised Edition 2006.
2. "Fundamentals of Database Systems", Elmasri, Navathe, Pearson Education, IVth Edition.
3. “An Introduction to Database Systems”, C. J. Date, Pearson Education.
4. “Introduction to Data Base Management”, Naveen Prakash, Tata McGraw Hill.
5. “Database Management Systems” , Ramakrishna, Gehrke; McGraw-Hill.
6. “Database Systems: A Practical Approach to design, Implementation and Management”, Thomas Connolly, Carolyn Begg; Third Edition, Pearson Education.
7. “A first course in Database Systems”, Jeffrey D. Ullman, Jennifer Windon, Pearson Education
8. “Data Management: databases and organization”, Richard T. Watson, Wiley Publication.
9. “Data Modeling Essentials”, Graeme C. Simxion, Dreamtech Publications.
10. MS-ACCESS Projects “Oracle 8i manuals”.

Title: Environnemental Science **Code: GE101**

L-T-P Scheme: 2-0-0 **Credit: 0**

Prerequisite: The students must be aware of basic Environmental Science upto class 12th. Basic knowledge of Environmental Science helps them to correlate in various division of Engineering during this course.

Objective:

The purpose behind this course is to make the students familiar with Environment (surrounding) and to understand the significance/importance of natural resource, biodiversity, environment pollution and impact of intervention of human being in the Ecosystem. This course is mandatory for all branches of the Engineering and Sciences.

Course Learning Outcomes:

Course Outcome	Description
CO1	The outline, outcomes and attributes provide students with learning experiences that help in learning the significance and importance of environment in their life.
CO2	Describe the real world problems, challenges with the suitable case study based on conservation (natural resource and biodiversity), ecosystem, socio-economic development and remedial measure of the various pollutions (air, water, soil, noise and radiation).
CO3	Develop in students the ability to apply the knowledge and skills they have acquired to the solution of specific theoretical and applied problems in their surrounding (the Environment).
CO4	Identify and use of various techniques for solving the Environmental Problems.
CO5	Apply filed visit and justification by using various analytical techniques.
CO6	Demonstrate students with the knowledge and skill base that would enable them to undertake further studies in the Environmental Science and related multidisciplinary areas that involve Environmental Science and help to develop a range of generic skills that are relevant to wage employment, self-employment and entrepreneurship.

Modules	**Description**	**No. of lectures**
Unit 1:	Introduction to Environmental Science: Multidisciplinary nature of environmental science; components of environment –atmosphere, hydrosphere, lithosphere and biosphere. Scope and importance; Concept of sustainability and sustainable development.	2
Unit 2:	Ecosystems: What is an ecosystem? Structure and function of ecosystem; Energy flow in an ecosystem: food chain, food web and ecological succession. Case studies of the following ecosystems: a) Forest ecosystem b) Grassland ecosystem c) Desert ecosystem d) Aquatic ecosystems (ponds, streams, lakes, rivers, oceans, estuaries)	4

Unit 3:	Natural Resources: Renewable and Non-renewable Resources • Land Resources and land use change; Land degradation, soil erosion and desertification. • Deforestation: Causes and impacts due to mining, dam building on environment, forests, biodiversity and tribal populations. • Water: Use and over-exploitation of surface and ground water, floods, droughts, conflicts over water (international & inter-state). • Heating of earth and circulation of air; air mass formation and precipitation. • Energy resources: Renewable and non-renewable energy sources, use of alternate energy sources, growing energy needs, case studies.	5
Unit 4:	Biodiversity and its conservation: Levels of biological diversity: genetic, species and ecosystem diversity; Biogeography zones of India; Biodiversity patterns and global biodiversity hot spots. • India as a mega-biodiversity nation; Endangered and endemic species of India. • Threats to biodiversity: habitat loss, poaching of wildlife, man-wildlife conflicts, biological invasions; Conservation of biodiversity: In-situ and Ex-situ Conservation of biodiversity. • Ecosystem and biodiversity services: Ecological, economic, social, ethical, aesthetic and Informational value.	4
Unit 5:	Environmental Pollution: Environmental pollution: types, causes, effects and controls; Air, water, soil, chemical and noise pollution. • Nuclear hazards and human health risks. • Solid waste management: Control measures of urban and industrial waste. • Pollution case studies.	5
Unit 6:	Environmental Policies & Practices: Climate change, global warming, ozone layer depletion, acid rain and impacts on human communities and agriculture.• Environment Laws : Environment Protection Act; Air (Prevention & Control of Pollution) Act; Water (Prevention and control of Pollution) Act; Wildlife Protection Act; Forest Conservation Act; International agreements; Montreal and Kyoto protocols and conservation on Biological Diversity (CBD). The Chemical Weapons Convention (CWC). • Nature reserves, tribal population and rights, and human, wildlife conflicts in Indian context.	4
Unit 7:	Human Communities and the Environment Human population and growth: Impacts on environment, human health and welfares. • Carbon foot-print. • Resettlement and rehabilitation of project affected persons; case studies. • Disaster management: floods, earthquakes, cyclones and landslides. • Environmental movements: Chipko, Silent valley, Bishnios of Rajasthan. • Environmental ethics: Role of Indian and other religions and cultures in environmental conservation. • Environmental communication and public awareness, case studies (e.g., CNG vehicles in Delhi).	4
Unit 8:	Field Work: Visit to a local area to document assets-river / forest / grassland /hill / mountain. polluted sites(Urban, rural ,industrial, agriculture), plants, insects, bird, Ecosystem (pond, river, hill slopes etc)	4
	Total	32

Teaching Methodology:

The core module Syllabus for Environment Science includes class room teaching and Field Work. The syllabus is divided into eight units covering lectures. The first seven units will cover 28 lectures, which are class room based to enhance knowledge skills and attitude to environment.

Unit eight is based on field activities which will be covered in 4 lecture hours and would provide student firsthand knowledge on various local environmental aspects. Field experience is one of the most effective learning tools for environmental concerns. This moves out of the scope of the text book mode of teaching into the realm of real learning in the field, where the teacher merely acts as a catalyst to interpret what the student observes or discovers in his/her own environment. Field studies are as essential as class work and form an irreplaceable synergistic tool in the entire learning process. Course material provided by UGC for class room teaching and field activities is utilized.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1 Unit 2 and Unit-3
Test-2	25 Marks	Based on Unit-4 & Unit-5 (70 %) and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-6 to Unit-7 and around 30% from coverage of Test-1 and Text-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Web Development (will be added from time to time): Digital copy will be available on the JUET server.

Text Book

1. Bharucha Erach, 2003. The Biodiversity of India, Mapin Publishing Pvt. Ltd, Ahmadabad – 380013, India.
2. De Anil Kumar, Environmental Chemistry, Wiley Eastern Ltd, 2007.
3. Agarwal KC, 2001. Environmental Biology, Nidhi Publishers Ltd. Bikaner.

Reference Book

1. Brunner RC, 1989, Hazardous Waste Incineration, McGraw Hill Inc. 480pgs.
2. Clark R B, Marine Pollution, Clanderson Press, Oxford (TB).2001.
3. Cunningham WP, Cooper TH, Gorhani E & Hepworth MT, 2001. Environmental Encyclopedia, Jaico Publishing House, Mumbai, 1196 pgs.
4. Gleick HP, 1993. Water in Crisis, Pacific Institute for Studies in Development, Environment and Security. Stockholm Environmental Institute, Oxford University Press, 473pgs.
5. Heywood VH, and Watson RT, 1995. Global Biodiversity Assessment. Cambridge University Press 1140pgs.
6. Jadhav H and Bhosale VM, 1995. Environmental Protection and Laws. Himalaya Publishing House, Delhi 284pgs.
7. Mckinney ML and Schoch RM, 1996. Environmental Science Systems and Solutions. Web enhanced edition, 639pgs.

Title of Course: Data Structures Lab **Course Code: CS203**
L-T-P Scheme: 0-0-2 **Credits: 1**

Scope and Objectives:

This course develop problem solving ability using programming, develop ability to express solutions to problems clearly and precisely, develop ability to design and analyze algorithms, introduce with fundamental data structures, develop ability to design and evaluate abstract data types and data structures.

Learning Outcome:

The students shall acquire the generic skills to design and implement data structures and related algorithms for a broad-based set of computing problems

18B11CI371: Data Structures Lab	
CO1	Define basic operations on linear data structures
CO2	Illustrate the efficiency of a data structures in terms of time and space complexity.
CO3	Apply the data structures solve the searching and sorting problems.
CO4	Utilize the knowledge of non-linear data structures in solving programming problems.
CO5	Analyze the data structures for their suitability on a given problem.
CO6	Design the systems, from concept to executable artefact using data structures techniques.

Course Description:

This course is intended to provide a thorough introduction to the use of data structures in programming. This course will cover the necessary mathematical background, but will assume the required programming experience.

Course Contents:

UNIT 1: Introduction to Data Structures, Algorithm and Complexity

Data structure overview, need of data structure and how to select relevant data structure for given problem, basic C data types and ADT.

Algorithm overview and its properties, problem analysis and construction of algorithm, difference between algorithm, program and software, algorithm analysis and complexity, asymptotic notations to represent the time complexity, Software Development Life Cycle (SDLC) phase

UNIT 2: Array

Overview, memory representation of 1D and 2D array, sparse matrix, operation supported by an array

Part 1: Searching

Linear search with illustration, analysis of linear search, binary search (iterative) and its analysis, binary search (recursive) and its analysis using recurrence relation, recurrence relation

Part 2: Sorting

Types of sorting algorithms, bubble sort, selection sort, insertion sort, quick sort, merge sort

UNIT 3: Linked List

Overview, types of linked list, linear linked list – overview, traversing, insertion, deletion, searching and reverse, doubly linked list – overview, traversing, insertion, deletion, circular linked list – overview, header linked list, applications of linked list

UNIT 4: Stack

Overview, stack implementation using stack and linked list, basic operations on stack using array and linked list – push, pop, dispose applications of stack – evaluation of mathematical expression, conversion of expression from one form to another (Polish Notation), Tower of Hanoi problem

UNIT 5: Queue

Overview, basic operations on queue – enqueue, dequeue, implementation of queue using array and linked list, types of queue - linear queue, circular queue, deque, priority queue, applications

UNIT 6: Tree

Tree definition and its terminology, representation of graph using array and linked list, tree traversals – preorder, inorder and postorder, binary search tree (BST) with insertion, deletion and searching operations, extended binary tree and its application in Huffman tree, threaded binary tree

UNIT 7: Graph

Introduction to graph, types of graph, traversal algorithms in graph – breadth first search, depth first search, spanning tree, minimum cost spanning tree - Kruskal's, Prim's.

Text Book:

T1: Sartaj Sahni, "Fundamentals of Data Structures", Tata Mc Graw Hill, New York
T2: Seymour Lipschutz., "Data Structures with C", Schaum's Outline Series
T3: Narasimha Karumanchi, "Data Structures and Algorithms" Made Easy

Reference Books:

R1: Corman et al: Introduction to Computer Algorithms
R2: Langsam, Augestein, Tenenbaum: Data Structures using C and C++
R3: Weiss: Data Structures and Algorithm Analysis in C/C++
R4: Samir K. Bandyopadhyay," Data Structures using C"
R5: Hopcraft, Ullman: Data Structures and Algorithms

Evaluation Scheme:

Component & Nature	**Marks**
Lab work	40
Lab record	15
Mid sem lab –Viva/Test	15
End sem lab – Viva/Test	15
Attendance & discipline in lab	15
Total	**100**

Title: Digital System & Microprocessors Lab **Code: EC206**
L-T-P scheme: 0-0-1 **Credit: 1**

Prerequisite: Students must have already studied the courses, "Digital Electronics and Microprocessor

Objective:

1. To learn and be able to implement the front-end and back-end digital electronics
2. To develop the abilities to call oneself full-stack microprocessor

Course Outcome	Description
CO1	Get familiar with basic of Digital Electronics
CO2	Understanding of logic gates and flip flops
CO3	Demonstration of combinational and sequential circuits
CO4	To understand the operation of ALU
CO5	Analyze the basic operations of 8085 microprocessor

Experiment No 1: Familiarization and Verification of logic functions of the TTL ICs.

Activity 1: Verification of AND gate using 7408 IC.
Activity 2: Verification of OR gate using 7432 IC.
Activity 3: Verification of NOT gate using 7404 IC.
Activity 4: Verification of NAND gate using 7400 IC.
Activity 5: Verification of NOR gate using 7402 IC.
Activity 6: Verification of XOR gate using 7486 IC.

Experiment No 2: Implementation of Combinational digital circuits using MSI Logic.

Activity 1: Combinational circuit-1
Activity 2: Combinational circuit-2

Experiment No 3: Implementation of Binary Adders and Subtractors.

Activity 1: Implementation of the Half-Adder.
Activity 2: Implementation of the Full-Adder using two Half-Adders.
Activity 3: Implementation of the Half-Subtractor.
Activity 4: Implementation of the Full-Subtractor using two Half-Subtractors.
Activity 5: Implementation of the 4-Bit Parallel Adder using ICs 7483.
Activity 6: Implementation of the 4-Bit Parallel Subtractor using IC 7483.

Experiment No 4: K-map and Boolean function simplification

Activity 1: Simplify the given digital circuit using K-map and verify the simplified function by implementing the given circuit and its simplified one.
Activity 2: Simplify the given functions whose minterm canonical formula is given. Implement the two functions with identical inputs and only use NAND gate ICs. Verify your result from the truth table.
Activity 3: Simplify the given Boolean function using minterms and maxterms. Implement both the simplified functions and verify that the functions are complement to each other. Construct the truth table as per your input/output behavior of the circuit.

Experiment No 5: Implementation of Multiplexer

Activity 1: Implementation of 2-to-1 Multiplexer using gates.
Activity 2: Implementation of 2-to-1 Multiplexer with enable/disable control signal.

Activity 3: Implementation of 2-to-1 Multiplexer using IC 74157.
Activity 4: Implementation of 4-to-1 Multiplexer using IC 74153.
Activity 5: Implementation of 8-to-1 Multiplexer using 4-to-1 MUX (IC 74153)

Experiment No 6: Use of Flip-Flop TTL IC in digital system.
Activity 1: Design and Implement NAND gated SR Latch
Activity 2: Design and Implement clocked RS Flip-Flop
Activity 3: Design and Implement D Flip-Flop using IC 7474.
Activity 4: Design and Implement JK Flip-Flop using IC 7476.
Activity 5: Design and Implement Master-Slave JK Flip-Flop.

Experiment No 7: Implementation of 4-Bit Binary Counter.
Activity 1: Implementation of 4-Bit Binary counter using 7493 IC .The clock signal to be giventhrough the pulsar and 1 Hz clock generator, and observe the output through LED.
Activity 2: Draw the waveform of the counter outputs Q_A, Q_B, Q_C and Q_D
Activity 3: Implementation of BCD counter using 7493 IC. Observe the output through sevensegment display.
Activity 4: Implementation of Mod-5 counter using 7493 IC.
Activity 5: Implementation of Mod-7 counter using 7493 IC.

Experiment No 8: Implementation of Shift Registers
Activity 1: Implementation of 4.bit Serial load parallel out (SIPO) shift register using 7474 IC.
Activity 2: Implementation of 4.bit parallel load serial out (PISO) shift register using 7474 IC.
Activity 3: Use of universal shift register IC 74194

Experiment No.9: Familiarization with 8085 microprocessor Kit
Activity 1: Draw and describe the each block of 8085 microprocessor kit
Activity 2: Practice the different command for Assembly Language Programming (ALP) of 8085 microprocessor

Experiment No.10: To perform loading and movement related instructions
Activity 1: Move the given data from accumulator to register
Activity 2: Load the content of memory location directly to the accumulator
Activity 3: Place the content of the memory location in register

Experiment No.11: **Aim: To carry out addition & subtraction operation.**
Activity 1: Perform the addition of given numbers
Activity 2: Addition of two 8- bit hexadecimal numbers
Activity 3: Addition of two 16- bit hexadecimal numbers
Activity 4: Perform the subtraction of given numbers
Activity 5: Subtraction of 16-bit hexadecimal numbers

Experiment No.12: To observe larger and smaller from given numbers
Activity 1: Find the larger and smaller number
Activity 2: Locate the largest number among the ten numbers
Activity 3: Locate the smallest number among the five numbers

Evaluation Scheme:

Exams		Marks	Coverage
P-1		15 Marks	**Based on Lab Exercises: 1-7**
P-2		15 Marks	**Based on Lab Exercises: 8-14**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Learning Resources:

Study material of Digital System & Microprocessors Lab (will be added time to time): Digital copy will be available on the JUET server

Text Book:

1. Fundamental of Digital Electronics And Microprocessors
2. Digital Electronics and Microprocessors

Web References:

1. http://www.becbapatla.ac.in/uploads/BCE1555920601838.pdf
2. https://et.charlotte.edu/about-us/facilities-equipment-and-infrastructure/eet-laboratories-and-associated-equipment/digital

Journals References:

1. Microprocessors and Microsystems - Journal - Elsevier
2. Microprocessors and digital ICs for motion control - IEEE Xplore
3. Journal of Microprocessor Engineering(STM Journals)

Title of Course: Database Systems Lab **Course Code: CI204**
L-T-P Scheme: 0-0-2 **Course Credit: 1**

Objectives: To develop the ability to design, implement and manipulate databases as well as to build Database management systems.

Learning Outcome

1. Ability to design systems by using ER Modeling.
2. Ability to develop skills of writing applications by using SQL.
3. Ability to understand query optimization techniques and transaction processing.

18B11CI373: Database Systems Lab	
CO1	Define basic requirement and operations of file based and database systems.
CO2	Illustrate the relational database design using data definition, data manipulation queries.
CO3	Develop the database using relational database query, Identify the suitable of the data structures as per the requirements.
CO4	Utilize the knowledge of structured query language to develop and deploy the database for real life based problems.
CO5	Develop the normalize database for their suitability on a given problem.
CO6	Design the database systems, from concept to executable transaction, concurrency and recovery control using the real time based problems in group project based task .

Course Contents:

- SQL queries for the creation of tables and insertion of values into tables.
- SQL queries for viewing all data and specific data corresponding to a particular row or column in a table.
- SQL queries for the updation, deletion and dropping of tables.
- SQL queries for aggregation, range finding etc on the tables.
- SQL queries for renaming, truncating and destroying the tables.
- SQL queries for the use of not null, group by, having clause.
- SQL queries for the computation done on the table data.
- Exercise on nested SQL queries and sub queries.
- Use of cursors, triggers, functions and writing pl/sql block.
- A brief idea about oracle report builder.

Evaluation scheme:

Exams		**Marks**	**Coverage**
P-1		15 Marks	**Based on Lab Exercises: 1-7**
P-2		15 Marks	**Based on Lab Exercises: 8-14**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Text Book

1. SQL, PL/SQL the Programming Language of Oracle, Ivan Bayross, 3rd edition.

Title of Course: UI/UX Lab **Course Code: CS205**
L-T-P Scheme: 0-0-2 **Course Credit: 1**

Objectives

By mastering fundamental UI/UX design principles, user-centered strategies, and integrating UX principles with accessibility standards, students will refine designs iteratively through user feedback and usability testing, ultimately evaluating UI/UX prototypes to deliver exceptional digital experiences.

Learning Outcomes

Course Outcome	Description
CO1	**Explain** fundamental principles of user interface (UI) design and improving user experience (UX).
CO2	**Describe** how user-centered design enhances UI/UX solutions by prioritizing user needs and preferences.
CO3	**Design** user interfaces that integrate UX principles, usability standards, and accessibility guidelines to ensure optimal user experience for all users.
CO4	**Analyze** user feedback and usability testing results to iterate and improve UI/UX designs.
CO5	**Assess** the effectiveness of UI/UX prototypes through user testing and iterative design processes

Course Content

Unit 1: Introduction to UI/UX Design: Overview of UI/UX design principles, Importance of user-centered design in UI/UX, Introduction to accessibility standards in UX design

Unit 2: Mastering UX Principles: Understanding user psychology and behavior, Principles of effective user interface design, integrating accessibility guidelines into UX design

Unit 3: User-Centered Design Strategies: Conducting user research: interviews and surveys, Creating user personas and scenarios, Ideation and prototyping techniques in UX design

Unit 4: Usability Testing and Iterative Design: Planning and conducting usability tests, Analyzing usability test results, Iterative design process based on user feedback

Unit 5: Evaluation and Optimization: Methods for evaluating UX prototypes, incorporating feedback to optimize UI/UX designs, Final assessment and refinement of digital experiences

Text books

1. The Design of Everyday Things – by Don Norman.

2. The Elements of User Experience: User-Centered Design for the Web- by Jesse James Garrett

Evaluation Scheme

Evaluations		Marks	Remarks
P-1		15 Marks	
P-2		15 Marks	
Continuous Evaluations	Viva	20 Marks	
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Discipline and Punctuality and Attendance	15 Marks	
Total		**100 Marks**	

Title of Course: Advance Programming Lab-I **Course Code: CS206**

L-T-P scheme: 0-0-2 **Course Credits: 1**

Prerequisite: No explicit prerequisite course work is required, but students are expected to have a fundamental understanding of basic computer principles and previous experience using a personal computer.

Objective: To emphasize object-oriented programming concepts and the design of algorithms and related data structures. Problem decomposition and principles of software engineering are stressed throughout the course. Advance aspects of programming may be taken care off through Python.

Learning Outcomes:

Course Outcome	Description
CO1	Installation and understanding features of Python.
CO2	Describe Python data types to handle programming problems
CO3	Develop understanding looping to handle new data types
CO4	Identify appropriate methods to solve challenging problems.
CO5	Apply programming knowledge to solve real world problems in the form of Project

Course Contents:

An Introduction to Python: Introductory Remarks about Python, Strengths and Weaknesses, A Brief History of Python, Python Versions, Installing Python, Environment Variables, Executing Python from the Command Line, IDLE, Editing Python Files, Getting Help, Dynamic Types, Python Reserved Words, Naming Conventions.

Basic Python Syntax: Introduction, Basic Syntax, Comments, String Values, String Operations, The format Method, String Slices, String Operators, Numeric Data Types, Conversions, Simple Input and Output, The print Function.

Language Components: Introduction, Control Flow and Syntax, Indenting, The if Statement, Relational Operators, Logical Operators, True or False, Bit Wise Operators, The while Loop, break and continue, The for Loop.

Collections: Introduction, Lists, Tuples, Sets, Dictionaries, Sorting Dictionaries, Copying Collections, Summary.

Functions: Introduction, Defining Your Own Functions, Parameters, Function Documentation, Keyword and Optional Parameters, Passing Collections to a Function, Variable Number of Arguments, Scope Functions- "First Class Citizens", Passing Functions to a Function, Mapping Functions in a Dictionary, Lambda, Closures.

Exceptions: Errors, Run Time Errors, The Exception Model, Exception Hierarchy, Handling Multiple, Exceptions, raise, assert, Writing Your Own Exception Classes.

Classes in Python: Classes in Python, Principles of Object Orientation, Creating Classes, Instance Methods, File Organization, Special Methods, Class Variables, Inheritance, Polymorphism, Type Identification, Custom Exception Classes, Class Documentation-pydoc.

GUI in Python: Introduction, Base window, Widgets, Functions, Lambda Functions, Geometry manager, Sqlite3 Backend Connectivity, Handling images.

Project: Based on Learning in this course with database connectivity.

Text Book

1. Programming Python /Mark Lutz.

Reference Books

1. Think Python / Allen B Downey
2. Python 101 / Dave Kuhlman

Evaluation scheme:

<table>
<tr><th colspan="2">Exams</th><th>Marks</th><th>Coverage</th></tr>
<tr><td colspan="2">P-1</td><td>15 Marks</td><td>Based on Lab Exercises: 1-7</td></tr>
<tr><td colspan="2">P-2</td><td>15 Marks</td><td>Based on Lab Exercises: 8-14</td></tr>
<tr><td rowspan="4">Day-to-Day Work</td><td>Viva</td><td>20 Marks</td><td rowspan="4">70 Marks</td></tr>
<tr><td>Demonstration</td><td>20 Marks</td></tr>
<tr><td>Lab Record</td><td>15 Marks</td></tr>
<tr><td>Attendance & Discipline</td><td>15 Marks</td></tr>
<tr><td colspan="2">Total</td><td colspan="2">100 Marks</td></tr>
</table>

4th Semester:

Title of Course: Software Engineering **Course Code** CS105
L-T-P Scheme: 3-0-0 **Credits: 3**

Pre-requisite: Good Knowledge of Computer Programming

Post Course:
Object Oriented Software Engineering, Software Quality Management Objective: To engineer good quality software from its specification

Learning Outcomes

Software Engineering	
Course Outcome	**Description**
CO1	Outline various software models with respect to their needs of the customer requirement and concepts of some modeling language.
CO2	Describe the real world problems using software engineering concepts and tools.
CO3	Develop the software design to meet customer expectations using modeling language.
CO4	Identify and use various cost estimation techniques used in software engineering project management.
CO5	Apply verification and validation techniques on a given software project.
CO6	Demonstrate deployment and basic maintenance skills.

Course Outline:
Interactive Systems, Usability, Introduction to software engineering, Software process models, PSP, TSP Requirement Engineering: Requirement Elicitation, Analysis, Specification, SRS, Formal system development techniques, Analysis and Modeling: Data modeling, Functional modeling, Software Architecture and Design: Data design, Architectural Design Process, SADT, OOAD, function-oriented design

UML: Use case diagram, State diagram, Activity Diagram, Class Diagram, Sequence diagram, Collaboration diagram, Deployment Diagram, Event trace diagram, Design Patterns: Structural Patterns, Behavioral Patterns, Creational Patterns

Software Estimation- Estimating Size, Effort and Cost: Metric for Analysis, Metric for Design, COCOMO model, Putnam Model etc., Implementation and Integration: Coding standard and practices, Top-Down and Bottom–up Approach, Verification and Validation,

Software Testing: Structural testing, functional Testing, Testing Strategies, Test Case design.

Software Maintenance: Types, Cost of Software, maintenance, Software Maintenance Models
CASE Tool Taxonomy: Business Process Engineering tool, Process modeling and management tool, project planning tool, requirement tracking tool, Metric and management tool, documentation tool, system software tool etc.Introduction to software engineering for web and mobile applications.

Teaching Methodology:
This course should be conducted in a highly interactive environment. Students will work on different software projects in small groups. Exercises shall almost exclusively consist of design work and the laboratory shall be a place to develop these designs using CASE tools. As part of lab work there shall be a project to build a specification and convert it into working software using Rational Unified Process. Also, there shall be a testing project. There is a self learning component that shall be announced.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1 & Unit-2
Test-2	25 Marks	Based on Unit-3 to Unit-4 and around 20% from coverage of Test-1
Test-3	35 Marks	Based on Unit-5 to Unit-6 and around 40% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Text Book

1. The Unified Modeling Language Users Guide: Grady Booch, James Rambaugh, Ivar Jacobson, Addision Wesley.
2. Douglas Bell, "Software Engineering for students: a programming approach", 4th Ed Pearson Education, 2005.
3. Dines, Bjorna "Software Engineering: abstraction and Modelling"Vol.1, 2006, Springer Verlag Berlin Heidelberg (206).
4. Cooling Jin, "Software Engineering for real time systems, Addison Wesley.
5. Khoshgoftaar, Taghi M. "Software Engineering with Computational Intelligence".
6. Sommerville, Ian, "Software Engineering", 8th Edition, Pearson Education Ltd.
7. Pressman S. Roger, "Software Engineering: A practitioner's Approach", 7th Edition, McGraw Hill.

Web References:

1. https://onlinecourses.nptel.ac.in/noc20_cs68/preview
2. https://online.visual-paradigm.com/
3. https://www.coursera.org/learn/introduction-to-software-engineering

Title of Course: **Algorithms and Problem Solving** **Course Code: CS106**
L-T Scheme: **3-1-0** **Course Credit**: **4**

Prerequisites:
Student must have already registered for "Introduction to Computer and programming" (07B11CI101), Data Structures (07B21CI102).

Objectives:
- Strengthen higher level cognitive Skills of analysis, creation and evaluation.
- Strengthen Ability of data abstraction and problem solving using computer
- Strengthen ability to express solution to problem clearly and precisely.
- Strengthen ability to design and evaluate ADTs, nonlinear temporary and persistent data structures and also related algorithms.
- Introduce students to some domain specific data structures and related algorithms in various domains.

Learning Outcomes:

Upon completion of the subject, students will be able to:

1 Get **familiar** with different basic concepts of algorithms and analyze the performance of algorithms.
2 Have a **good grounding** of advance data structures like R-B Tree, M – way tree, models and IDEs.
3 Get to **learn** about various algorithm design techniques for developing algorithms.

4 Possess **demonstrative skills** in solving optimization problems.
5 Be able to **design, develop algorithms,** and employ appropriate data structures for solving real world computing problems efficiently.

Course Content:
Analysis of algorithm: Asymptotic Notation, Sorting and merging Algorithm
Tree and related data Structure: Heap, Priority Queues, B+ Tree, AVL, Splay Tree, Red-Black Tree, Threaded Tree
Files: Classification, Record Organization, Retrieval System, External Sorting
Set, Dictionary: Design, Analysis, integration and applications
Fundamental techniques: Divide and Conquer method, Dynamic Programming, Introduction to Greedy Method
Hashing: technique, collision resolution and analysis
Text Processing: String operation, pattern matching algorithm, tries, text compression, text similarity testing.

Teaching Methodology:

The Course will use the mixed technique of interactive lectures, guided case studies, literature survey, regular assignments and project work. In addition to the material covered in the class, student will be required to explore study, evaluate present and implement domain specific data structure in different domain. Teaching in this course is designed to engage the student in active and experimental learning by taking a problem solving and design oriented approach with special emphasis on real world applications. Lectures will be highly interactive and work oriented. Student will have to work individually as well as in groups inside as well as outside the class. Students are expected to carry out a lot of design and

programming oriented project work. Each student is expected to write minimum 3000 lines of documented program code as part of this course. Students are encouraged to learned use toolkits like STL for project implementation. Each student is also expected to do literature survey making use of the library and web resources (including digital library) to identify ,understand ,summarize and present at least one research paper on science and application of non-linear data structure and algorithms.

Evaluation Scheme (Theory):

Evaluation Scheme is designed to promote and test higher level thinking skills and de-emphasis rote learning through holistic and continuous evaluation. Written exam will be designed and conducted as open Book(s), open notes tests. One of the minor tests may me designed and conducted as a take home test. Evaluation scheme will have following components

Test-1	15 Marks
Test-2	25 Marks
Test-3	35 Marks
Home assignment /Quizzes	10 Marks
Tutorial/Problem solving session	10 Marks
Attendance	05 Marks
Total	**100** Marks

Text book

T1: Thomas H., Coremen: Introduction to algorithm, the Massachusetts institute of Technology, Cambridge, Massachusetts.

Reference Books:

1. Aho, Hopcraft, Ullman: Data Structure and Algorithms
2. Kruse, Tonso, Leung: Data Structure and program Design in C
3. Sahni: Data structure and algorithm and application in C++
4. Weiss: Data Structure and Algorithm analysis in C/C++

Course Name: Computer Organization and Architecture Course **Code: CS107**
L-T-P scheme: 3-1-0 **Credits: 4**

Pre-requisite: Digital System and Microprocessors

18B11CI414: Computer Organization & Architecture	
Course Outcome	**Description**
CO1	Develop the understanding of data representation and digital logic circuits used in the computer system.
CO2	Concepts of Register Transfer Language (RTL) to design data transfer bus, combinational and sequential logic circuits.
CO3	Understand the programming of basic computer system using machine language, assembly language and microinstructions.
CO4	Describe the various architectures of CPU and their designing concepts.
CO5	Memory hierarchy, cache memory, virtual memory and their working principle/performance.

Course Contents:

Unit-1: Digital Logic Circuits - Logic Gates, Boolean Algebra, Map Specification, Combinational Circuits, Flip-Flops, Sequential Circuits, Memory Components, Integrated Circuits. Data Representation - Data Types, Complements, Fixed Point Representations Floating Point Representations, Other Binary Codes, Error Detection Codes.

Unit-2: Register Transfer and Micro operations – Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro operations, Logic Micro operations, Shift Micro operations, Arithmetic Logic Shift Unit. Basic Computer Organization And Design- Instruction Codes, Computer Registers, Computer Instructions, Timing And Control, Instruction Cycle, Memory Reference Instructions, Input-Output And Interrupt, Complete Computer Description, Design Of Basic Computer, Design Of Accumulator Logic.

Unit-3: Programming The Basic Computer - Introduction to Machine Language, Assembly Language, Assembler, Program Loops, Programming Arithmetic And Logic Operations Subroutines, Input-Output Programming. Micro programmed Control-Control Memory, Sequencing Microinstructions, Micro program Example, Design Of Control Unit, Microinstruction Format.

Unit-4: Central Processing Unit – Introduction To CPU, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer And Manipulation, Program Control, Reduced Instruction Set Computer. Pipelining and Vector Processing - Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline.

Unit-5: Computer Arithmetic – Introduction to Computer Arithmetic, Addition and Subtraction, Multiplication algorithms, Division algorithms, floating point arithmetic Operations, Decimal Arithmetic Unit, Decimal Arithmetic Operations. Input-Output Organization - Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes Of Transfer, Priority Interrupt, Direct Memory Access, Input-Output Processor, Serial Communication.

Unit-6: Memory Organization - Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory, Memory Management Hardware. Multiprocessors-Characteristics of Multiprocessors.

Text Books

1. "Computer System Architecture", M. Morris Mano, Third Edition.
2. "Computer Organization and Architecture", William Stalling, Tenth Edition.

Other References:

1. Yu-cheng Liu,Glenn A. Gibson , "The 8086/8088 Family Architecture, Programming & design", Second Edition, PHI.
2. Douglas Hall, "Microprocessors & Interfacing, Programming & Hardware",2nd Edn. Tata McGraw Hill.
3. Kenneth Ayala **"**The 8086 microprocessor programming and Interfacing the PC".
4. Tom Shanley, Protected Mode Software Architecture, Addison-Wesley (1996), ISBN 0-201-55447-X .

Resources

Lecture presentations, assignments and practicals, will be posted on the student resource from time to time. In addition following additional online/downloadable resources will be useful.

- NPTEL Course: Computer architecture and organization, IIT Kharagpur by Prof. Indranil Sengupta, Prof. Kamalika Datta, https://nptel.ac.in/courses/106105163
- Official IA-32 Programmer Reference Manuals online at http://developer.intel.com/design/Pentium4/documentation.htm
- Professor Ralf Brown's Interrupt List online at http://www.ctyme.com/rbrown.htm
- Homepage for H. Peter Anvin's SYSLINUX Project online at http://syslinux.zytor.com/
- Online article: The GNU GRUB Boot Loader by Jaswinder Singh Kohli (Linux Gazette #64, 2001)
- Official Data Sheet for the Intel 8259A Programmable Interrupt Controller (.pdf file-format)

Evaluation Scheme:

Test-1	15 marks
Test-2	25 marks
Test-3	35 marks
Assignments	10 marks
Tutorial	5 marks
Quizzes	5 marks
Total	**100 marks**

Title of Course: Machine Learning **Course Code: CS108**
L-T-P scheme: 3-0-0 **Credit: 3**

Prerequisite: The mathematical tools needed for the course will be covered in some classes in the first week of the course.

Objective:
1. To learn and be able to implement the basic statistical techniques in the areas of interest.
2. To develop the abilities to apply the basic Machine Learning algorithms and interpret their results.

Learning Outcomes:

At the end of the course, students:

1. Get familiar with the fundamental methods at the core of modern machine learning.
2. Have a good grounding of the essential algorithms for supervised and unsupervised learning
3. Possess demonstrative skills in using and applying Machine Learning.
4. Work as a team on a project.

Course Outcome	**Description**
CO1	List various approaches of Machine Learning.
CO2	Describe machine learning algorithms to solve the real world problems
CO3	Develop Hypothesis and machine learning models
CO4	Identify appropriate models for solving machine learning problems.
CO5	Apply learning techniques to solve real world machine learning problems
CO6	Evaluate and interpret the results of the algorithms.

Course Content:

Unit-I: Introduction to machine learning, supervised and unsupervised machine learning, Applications of AI and machine learning , Linear Algebra, Matrices, Multi-Variable Calculus and Vectors, Mean, Median, mode, Dispersion.

Unit-II: Probability, Probability Distributions, and Central Limit Theorem.
Hypothesis Testing: The what, why and how of Hypothesis Testing are covered in this module. P-Value, different types of tests and implementation in Python.
Exploratory Data Analysis: EDA brings out the information from the Data. This module covers Data Cleaning, Univariate/ Bivariate analysis.

Unit-III: Linear Regression: Simple and Multiple, Issues in Regression like Collinearity. Project on Linear Regression. Logistic Regression Univariate and Multivariate Logistic Regression for classification in ML, Implementation in R/Python, Naive Bayes Classification. Bias-Variance Tradeoff, Evaluation metrics: Confusion Matrix, F1 Score, Root Mean Squared Error.

Unit-IV: Decision Tree, Random Forest, SVM, Validation Techniques: Leave one out cross-validation, K-fold cross-validation, Stratified k-fold cross-validation.

Unit-V: K-Means clustering, Introduction to Neural Networks, Convolutional Neural Network.

Teaching Methodology:

This course is introduced to help students understand the discipline of Machine Learning. The programming tool used to teach this course are R and Python. Starting from the basic mathematical tools, the student will slowly be exposed to inferential statistics, and later to Machine Learning Algorithms. This theory course is well complemented by a laboratory course under the name Machine Learning Lab in the same semester that helps a student learn with hand-on experience.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Based on Unit-1 & Unit-2
Test-2	25 Marks	Based on Unit-3 & Unit-4 and around 20-30% from coverage till Test-1
Test-3	35 Marks	Based on Unit-5 to Unit-6 and around 30% from coverage till Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Machine Learning (will be added from time to time): Digital copy will be available on the JUET server.

Text Book:

- Hastie, Tibshirani and Friedman. Elements of statistical learning.

Reference Material:

- L. Rosasco. Introductory Machine Learning Notes.
- Larry Wasserman. Clustering chapter

Title: Algorithms Lab **Code: CS207**
L-T-P scheme: 0-0-2 **Credit: 1**

Prerequisite: Experience in programming is desirable. Student must have already registered for "Software Development Lab" (18B17CI171) and "Data Structures lab" (18B17CI371).

Objective:

3. To provide exposure to problem-solving through programming.
4. Strengthen higher level cognitive Skills of analysis of problem, creation of solution and evaluation of performance.
5. Strengthen Ability of data abstraction and problem solving using computer
6. Strengthen ability to express solution to problem clearly and precisely.
7. Strengthen ability to design and evaluate ADTs, nonlinear temporary and persistent data structures and also related algorithms.
8. Introduce students to some domain specific data structures and related algorithms in various domains.

Learning Outcomes:

Course Outcome	Description
CO1	Design new algorithms, prove them correct, and analyze their asymptotic and absolute runtime and memory demands.
CO2	Find an algorithm to solve the problem (create) and prove that the algorithm solves the problem correctly (validate).
CO3	Understand the mathematical criterion for deciding whether an algorithm is efficient, and know many practically important problems that do not admit any efficient algorithms.
CO4	Apply classical sorting, searching, optimization and graph algorithms.
CO5	Understand basic techniques for designing algorithms, including the techniques of Recursion, Divide-and-Conquer, Greedy Algorithms and Dynamic Programming

Course Content:

The following assignments will be carried out in synchronization with the theory classes.

Unit-1: Development of programs including analysis of algorithm Asymptotic Notation, Sorting and merging Algorithm.

Unit-II: Programs using Heap, Priority Queues, B-Tree, AVL, Splay Tree, Red-Black Tree, Threaded Tree.
Unit-III: Programs using Classification, Record Organization, and Retrieval System of files External Sorting. Design, Analysis, integration of set & dictionary, collision resolution and analysis
Unit-IV: Programs using Divide and Conquer method, Dynamic programming, Introduction to Greedy Method.

Unit-V: Program using String operation, pattern matching algorithm, tries, text compression, text similarity testing application.

Units to Lab Mapping:

Unit	Labs
I	1, 2, 3
II	4, 5
III	6, 7, 8
IV	9, 10, 11
V	12, 13, 14

Teaching Methodology:

This course is introduced to help students understand the designing and analysis of algorithm. Any (C, C++, JAV etc) programming language used to implement algorithms. Starting from the programming environment setup, the student will slowly be exposed to program designing and later to complexity analysis fundamentals. The entire course is broken down into five separate units, from fundamentals of algorithms to some complex algorithms designing methodology like Dynamic Programming Greedy Techniques etc.

Evaluation Scheme:

<table>
<tr><th colspan="2">Exams</th><th>Marks</th><th>Coverage</th></tr>
<tr><td colspan="2">P-1</td><td>15 Marks</td><td>Based on Lab Exercises: 1-6</td></tr>
<tr><td colspan="2">P-2</td><td>15 Marks</td><td>Based on Lab Exercises: 7-13</td></tr>
<tr><td rowspan="4">Day-to-Day Work</td><td>Viva</td><td>20 Marks</td><td rowspan="4">70 Marks</td></tr>
<tr><td>Demonstration</td><td>20 Marks</td></tr>
<tr><td>Lab Record</td><td>15 Marks</td></tr>
<tr><td>Attendance & Discipline</td><td>15 Marks</td></tr>
<tr><td colspan="2">Total</td><td colspan="2">100 Marks</td></tr>
</table>

Learning Resources:

Study material of Algorithms Lab (will be added time to time): Digital copy will be available on the JUET server.

Text Book:

1. Thomas H., Coremen: Introduction to algorithm, the Massachusetts institute of Technology, Cambridge, Massachusetts.

Reference Books/Material:

1. Aho, Hopcraft, Ullman: Data Structure and Algorithms
2. Kruse, Tonso, Leung: Data Structure and program Design in C
3. Sahani: Data structure and algorithm and application in C++
4. Weiss: Data Structure and Algorithm analysis in C/C++

Online Courses:
NPTEL-Algorithms and Problem Solving: https://nptel.ac.in/courses/106/105/106105164/

Videos Available on YouTube:
https://www.youtube.com/watch?v=OQ5jsbhAv_M

https://www.youtube.com/watch?v=huQojf2tevI
https://www.youtube.com/watch?v=sSno9rV8Rhg

Website

- https://www.geeksforgeeks.org
- https://www.indiabix.com
- https://www.includehelp.com
- https://www.tutorialspoint.com
- https://www.sanfoundry.com
- https://www.programiz.com

Coding Platforms

- https://www.codechef.com
- https://www.hackerrank.com
- https://www.interviewbit.com
- https://www.spoj.com
- https://www.hackerearth.com
- https://leetcode.com

Integrated Development Environment

- Turbo C++
- Dev-c++
- Code::Block

Title of Course: Computer Organization and Architecture Lab **Course Code: CS208**
L-T-P scheme: 0-0-2 **Credit: 1**

Prerequisite: Students must have knowledge of Digital Systems and Microprocessors (DSM) subject.

Objective:

1. To acquire the generic hardware development skill through various stages of designing.
2. To design and verify computer system digital circuits in VHDL.
3. To ensure the quality of hardware through various levels of verifications with Xilinx software and ISim simulator.

Learning Outcomes:

Course Outcome	Description
CO1	Designing of basic building blocks of a computer system.
CO2	Implementation of basic adder-subtractor units.
CO3	Learn to design the ALU of a computer system.
CO4	Understand the designing of computer data bus architecture.
CO5	Memory (RAM/ROM) system designing.
CO6	Designing of sequential logic circuits for a computer system.

Course Content:
Unit-1; Introduction to VHDL and Xilinx ISE Software.
Unit-2: Design of All-in-One logic gate circuits.
Unit-3: Design of 4-bit adder-subtractor circuits.
Unit-4: Design of combinational logic circuits.
Unit-5: Design of multiplexer based N-bit common bus system, logic system, and shift system.
Unit-6: Design of Arithmetic Logic Shift Unit (ALU).
Unit-7: Design of registers and counters.
Unit-8: Design of ROM and RAM.

Evaluation Scheme:

Exams		**Marks**	**Coverage**
P-1		15 Marks	**Based on Lab Exercises: 1-4**
P-2		15 Marks	**Based on Lab Exercises: 5-8**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Learning Resources:

Soft copies of study material and lab exercises of Computer Organization and Architecture Lab are made available on the JUET server.

Text Book

1. "VHDL Primer"; J. Bhasker , 3 rd Edition.
2. "Computer System Architecture" by M Morris Mano, Third Edition.

Other References:

0. The student's guide to VHDL"; Peter. J. Ashenden, 2nd Edition.
1. Computer System Organization and Architecture: Designing for Performance" by W Stallings, Seventh Edition, Prentice Hall, 2006. ISBN: 0-13-185644-8.

Resources

Lecture presentations, assignments and practicals, will be posted on the student resource from time to time. In addition following additional online/downloadable resources will be useful.

- Xilinx Documentation Portal: https://docs.xilinx.com/r/en-US/xilinx-documentation-portal
- NPTEL Course: Computer architecture and organization, IIT Kharagpur by Prof. Indranil Sengupta, Prof. Kamalika Datta, https://nptel.ac.in/courses/106105163

Virtual Lab: http://vlabs.iitkgp.ac.in/coa

Title of Course: Machine Learning Lab **Course Code: CS209**
L-T-P scheme: 0-0-2 **Credit: 1**

Prerequisite: The mathematical tools needed for the course will be covered in some classes in the first week of the course.

Objective:

1. To learn and be able to implement the basic statistical techniques in the areas of interests.
2. To develop the abilities to apply the basic Machine Learning algorithms and interpret their results.

Learning Outcomes:
At the end of the course, students:

Course Outcome	**Description**
CO1	Get familiar with the fundamental methods at the core of modern machine learning.
CO2	Have a good grounding of the essential algorithms for supervised and unsupervised learning
CO3	Possess demonstrative skills in using and applying Machine Learning.
CO4	Work as a team on a project.

Course Content:

Unit-I: Introduction to machine learning, supervised and unsupervised machine learning, Applications of AI and machine learning , Linear Algebra, Matrices, Multi-Variable Calculus and Vectors, Mean, Median, mode, Dispersion.

Unit-II: Probability, Probability Distributions, and Central Limit Theorem.
Hypothesis Testing: The what, why and how of Hypothesis Testing are covered in this module. P-Value, different types of tests and implementation in Python.
Exploratory Data Analysis: EDA brings out the information from the Data. This module covers Data Cleaning, Univariate/ Bivariate analysis.

Unit-III: Linear Regression: Simple and Multiple, Issues in Regression like Collinearity. Project on Linear Regression. Logistic Regression Univariate and Multivariate Logistic Regression for classification in ML, Implementation in R/Python, Naive Bayes Classification. Bias-Variance Tradeoff, Evaluation metrics: Confusion Matrix, F1 Score, Root Mean Squared Error.

Unit-IV: Decision Tree, Random Forest, SVM, Validation Techniques: Leave one out cross-validation, K-fold cross-validation, Stratified k-fold cross-validation.

Unit-V: K-Means clustering, Introduction to Neural Networks, Convolutional Neural Network.

Teaching Methodology:

This course is introduced to help students understand the discipline of Machine Learning. The programming tool used to teach this course are R and Python. Starting from the basic mathematical tools, the student will slowly be exposed to inferential statistics, and later to Machine Learning Algorithms. This theory course is well complemented by a laboratory course under the name Machine Learning Lab in the same semester that helps a student learn with hand-on experience.

Evaluation Scheme:

Evaluations		**Marks**	**Remarks**
P-1		15 Marks	
P-2		15 Marks	
Continuous Evaluations	Viva	20 Marks	
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Discipline and Punctuality and Attendance	15 Marks	
Total		**100 Marks**	

Learning Resources:

Lab exercises and lecture slides on Machine Learning (will be added from time to time): Digital copy will be available on the JUET server.

Text Book:

- Hastie, Tibshirani and Friedman. Elements of statistical learning.

Reference Material:

- L. Rosasco. Introductory Machine Learning Notes.
- Larry Wasserman. Clustering chapter

Title of Course: Mobile and Application Development Lab **Course Code:** CS210

L-T-P scheme: 0**-0-2** **Credit: 1**

Prerequisite: Students must have already registered for the course, "***Introduction to Computers and Programming***" and "**Object Oriented Programming**".

Objective:

1. To learn and be able to implement different mobile-technologies.
2. To develop the abilities to call oneself mobile application developer.

Learning Outcomes:

At the end of the course, a student will:

1. Get familiar with different approaches to mobile application development.
2. Get to learn about application marketing.
3. Have a good grounding of mobile application development requirements, models and IDEs.
4. Possess demonstrative skills in building native applications.
5. Be able to design and develop cross-platform applications.
6. Learn to work in a team on a project.

Course Content:

Part-1: Orientation and Fundamentals of Development

Unit-1 Mobile applications and different approaches to mobile application development. Java features and review of Object Oriented Programming fundamentals.

Part-2: Android Studio and Basic Development Skills

Unit-2 Installing and getting accustomed to the android studio environment. Using activities and views. Working on different views like TextViews, ImageViews etc. Creating simple applications using basic view types.

Unit-3 Using animations, audio and video. Advanced android features like list views, Exception handling, Timers in androids, Advanced String manipulations.

Part-3: Serious Development

Unit-4 Maps and GeoLocation, Storing data permanently, Alert dialogs, SQLite databases, Advanced SQLite, Webviews.

Unit-5 Submitting app to distribution channels, marketing mobile app, Mobile App development models.

Part-4: Working in a team and Cross Platform Development

Unit-6 Using Git, Common Git commands, Project Development, Cross Platform Development using Flutter, Coding using Dart, MVC design pattern, Networking, Data storage, Authentication, State Management.

Teaching Methodology:

This course is introduced to help students transition from a regular developer to a mobile app developer. Starting from the basics, the student will slowly progress to become to other aspects of development including database, version control and other essential technologies that are

helpful for a developer. The entire course is broken down into four separate parts: Orientation and Fundamentals of Development, Android Studio and Basic Development Skills, Serious Development, and Working in a team and Cross Platform Development. Each section includes multiple technologies to help a student gain more experience as a developer. This lab course is well complemented by a lecture in the same semester that helps a student learn and discuss the technical details of the underlying technologies.

Evaluation Scheme:

Exams		**Marks**	**Coverage**
P-1		15 Marks	**Based on Lab Exercises: 1-7**
P-2		15 Marks	**Based on Lab Exercises: 8-14**
Day-to-Day Work	Viva	20 Marks	**70 Marks**
	Demonstration	20 Marks	
	Lab Record	15 Marks	
	Attendance & Discipline	15 Marks	
Total		**100 Marks**	

Learning Resources:
Tutorials and lecture slides on Mobile Development (will be added from time to time): Digital copy will be available on the JUET server.

Books:

Text Book

i. Hello, Android (3rd edition): Introducing Google's Mobile Development Platform by Ed Burnette ISBN: 978-1-93435-656-2

ii. Android Programming for Beginners: Build in-depth, full-featured Android 9 Pie apps starting from zero programming experience, 2nd Edition by John Horton ISBN: 978-1789538502

iii. Head First Android Development: A Brain-Friendly Guide 1st Edition by Dawn Griffiths, David Griffiths. ISBN: 978-1449362188

Reference Books

1. Android Programming: The Big Nerd Ranch Guide (3rd Edition) (Big Nerd Ranch Guides) 3rd Edition by Bill Phillips, Chris Stewart, Kristin Marsicano ISBN: 978-0134706054
2. The Busy Coder's Guide to Android Development Version 8.0 by Mark M Murphy (O nline Book)

Web References:

1. https://developer.android.com
2. https://www.androidauthority.com
3. https://www.vogella.com

Journals:

1. International Journal of Interactive Mobile Technologies (iJIM)
2. ACM Transactions on the Information Systems (TOIC).
3. International Journal of Modern Computer Science (IJMCS)
4. ACM Transactions on Internet Technology (TOIT).

5th Semester

Title: Probability Theory and Random Processes **Code: MA106**

L-T-P scheme: 3-1-0 **Credit: 4**

Prerequisite: Students must have already studied course, "Mathematics-I" and should have the Knowledge of Differential & Integral Calculus.

Objective: Objective of this course is to provide a foundation in the theory and applications of probability and stochastic processes and an understanding of the mathematical techniques relating to random processes in the areas of signal processing, detection, estimation, and communication. Topics include the axioms of probability, random variables, and distribution functions; functions and sequences of random variables; stochastic processes; and representations of random processes.

Learning Outcomes:

Course Outcome	Description
CO1	Construct sample spaces of random experiments; identify and specify events, and perform set operations on events; compute probabilities by counting; evaluate conditional probability, and apply Bayes' theorem to simple situations.
CO2	Express random variables by using CDFs, PMFs; calculate moments related to random variables; understand the concept of inequalities and probabilistic limits. Understand the axiomatic approach of probability theory and intrinsic need of (functions of) random variables for the analysis of random phenomena.
CO3	Compute probability distributions and correlation measures of bivariate random variables; obtain marginal and conditional distributions of random variables; find probabilities for outcomes of various events related to an uncertain phenomenon using appropriate probability distributions as models.
CO4	Conduct hypotheses tests concerning population parameters based on sample data; perform and interpret chi-square test of goodness-of-fit and test of independence; find the equation of regression line and second degree curve, and to predict the value of one variable based on the value of the other variable.
CO5	Identify and classify random processes and determine covariance and spectral density of stationary and ergodic random processes; demonstrate specific applications to Gaussian process.
CO6	Students are able to provide the theories associated with the random variable and random process. The course particularly provides the student with an ability to apply to real-world problems in the communication and physical systems.

Course Contents:

Unit-1: Random experiments, sample space and events. Three basic approaches to probability, conditional probability, total probability theorem, Bayes' theorem of Probability of causes, Bayes' theorem of future events, total independence, mutual independence and pair wise independence.

Unit-2: One dimensional random variables(discrete and continuous) and their distributions, bivariate distributions, joint, marginal and conditional distributions, characteristic

function.

Unit-3: Covariance and correlation of random variables. Some special probability distributions: Binomial, Poisson, probability distributions. Negative Binomial,Geometric and Normal probability distributions. Fitting of probability distributions.

Unit-4: Concept of reliability: Reliability function, Hazard rate function, Mean time to failure, cumulative and average failure rate, Conditional reliability and failure rates, residual MTTF, some special failure rate distributions- exponential distribution and the Weibull distribution, reliability of systems- series configuration and some deductions, parallel-series configuration, series -parallel configuration.

Unit-5: Introduction and description of random processes, average values of random processes, stationary processes and computation of their averages, autocorrelation function and its properties, Cross correlation and its properties. Power spectral density function and its properties. Ergodicity of a random process, Poisson processes.

Teaching Methodology:

The course will be covered through lectures supported by tutorials. Apart from the discussions on the topics covered in the lectures assignments/ quizzes in the form of questions will also be given.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Based on Unit-1
Test-2	25 Marks	Based on Unit-2 & Unit-3 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-4 to Unit-5 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Probability Theory and Random Processes (will be added from time to time): Digital copy will be available on the JUET server.

Text books:

1. T. Veerarajan ,Probability, Statistics and Random Processes, Tata McGraw Hill.
2. J.J. Aunon & V. Chandrasekhar, Introduction to Probability and Random Processes, Mc-Graw Hill International Ed.
3. A. Papoulis & S.U. Pillai, Probability, Random Varibles and Stochastic Processes, Mc-Graw Hill.
4. H. Stark, and J.M. Woods, Probability and Random Processes with Applications to Signal Processing, Pearson Education.

Title of Course: Computer Networks **Course Code: CS109**

L-T-P Scheme: 3-0-0 **Credit: 3**

Objective:

The objective of this course is to build basic concepts of Computer network established for the data communication. This course also aims to provide the fundamental concepts in the design and implementation of networks, their protocols and its applications.

Learning Outcomes:

Computer Networks	
Course Outcome	**Description**
CO1	Outline basics to advanced concepts and techniques of Computer networks.
CO2	Describe problem solving approaches as applied in Data communication networking areas.
CO3	Analyse performance of basic communication networks using both analytical and simulation techniques.
CO4	Develop the Computer network design techniques and practical implementation issues.
CO5	Understand the basic properties of internet and data traffic properties.
CO6	Apply verification and validation techniques on a given software project.
CO7	Demonstrate deployment and basic maintenance skills.

Course Content:

UNIT - I Network hardware, Network software, OSI, TCP/IP Reference models, Example Networks: ARPANET, Internet. Physical Layer: Guided Transmission media: twisted pairs, coaxial cable, fiber optics, Wireless transmission.

UNIT - II Data link layer: Design issues, framing, Error detection and correction. Elementary data link protocols: simplex protocol, A simplex stop and wait protocol for an error-free channel, A simplex stop and wait protocol for noisy channel. Sliding Window protocols: A one-bit sliding window protocol, A protocol using Go-Back-N, A protocol using Selective Repeat, Example data link protocols. Medium Access sub layer: The channel allocation problem, Multiple access protocols: ALOHA, Carrier sense multiple access protocols, collision free protocols. Wireless LANs, Data link layer switching.

UNIT - III Network Layer: Design issues, Routing algorithms: shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing, Congestion Control Algorithms, Quality of Service, Internetworking, the Network layer in the internet.

UNIT - IV Transport Layer: Transport Services, Elements of Transport protocols, Connection management, Transmission Control Protocol and User datagram protocols.

UNIT - V Application Layer –Domain name system, Simple Network Management Protocol, Electronic Mail; the World WEB, HTTP, File transfer protocol, Security related issues.

Teaching Methodology

This course will help the students to facilitate interaction and information transfer over large distances. With internet, computer and telephone networks, buisenesses can allocate their resources efficiently. The Students will be able to learn basic concepts of computer network, its working principle & operation of Internet and Intranet. They will also learn the working principle of operation of LAN, WAN, MAN, congestion in the network and network management.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Based on Unit-1, Unit-2
Test-2	25 Marks	Based on Unit-2, 3 & Unit-4 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-4 and Unit-5 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Telecommunication networks (will be added from time to time):

Text Books:

1. A.S. Tennenbaum, Computer Networks, PHI
2. W. Stallings, Data & Computer Communication, PHI
3. Forouzen, Behrouz A.Fegan, Sophia Chung Data Communications and Networking, TMH

Reference Books:

1. Carne, E. Bryan Professional's Guide to Data Communication in a TCP/IP World Artech House, London, 2004
2. Young, Margret Levine Internet: The Complete Reference, Tata McGraw Hill, New Delhi, 2002

Web References:

1. **www.britannica.com**
2. **www.vssut.ac.in**

Journals References:

1. International Journal on Advances in Telecommunications
2. Journal of Network and Computer applications- Elsevier
3. IEEE transactions on networking
4. ACM Journals on networking

Title: Theory of Computation **Code:** CS110

L-T-P scheme: 3-1-0 **Credit: 4**

Prerequisite:

Students must have already studied for the course Set algebra, elementary formal logic, constructing proofs, recurrence relations.

Objective:

1. To give an overview of the theoretical foundations of computer science from the perspective of formal languages
2. To illustrate finite state machines to solve problems in computing
3. To explain the hierarchy of problems arising in the computer sciences.
4. To familiarize Regular grammars, context frees grammar.

Learning Outcomes:

Course Outcome	Description
CO1	Students will demonstrate knowledge of basic mathematical models of computation and describe how they relate to formal languages.
CO2	To Design Finite Automata's for different Regular Expressions and Languages
CO3	To Construct grammar for various languages and applying normal forms and push down automata
CO4	To solve various problems of Turing Machines and types of TM

Course Content:

UNIT – I

Mathematical Concepts: Review definitions and notations for sets, relations and functions. Basic concepts and definitions Set operations; partition of a set, Equivalence relations; Properties on relation on set; Proving Equivalences about Sets. Central concepts of Automata Theory.

UNIT – II

FINITE AUTOMATA (FA): Introduction, Deterministic Finite Automata (DFA) -Formal definition, simpler notations (state transition diagram, transition table), language of a DFA. Nondeterministic Finite Automata (NFA)- Definition of NFA, language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machines) and Inter conversion.

UNIT - III

REGULAR EXPRESSIONS (RE): Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions- Converting from DFA's to Regular Expressions, Converting Regular Expressions to Automata, applications of Regular Expressions.

REGULAR GRAMMARS: Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular -Pumping lemma, applications, Closure properties of regular languages.

UNIT - IV

CONTEXT FREE GRAMMER (CFG): Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG's, Minimization of CFG's, CNF, GNF, Pumping Lemma for CFL's, Enumeration of Properties of CFL (Proof's omitted).

UNIT – V

PUSHDOWN AUTOMATA: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA.

UNIT VI

TURING MACHINES (TM): Formal definition and behaviour, Languages of a TM, TM as accepters,

and TM as a computer of integer functions, Types of TMs.

RECURSIVE AND RECURSIVELY ENUMERABLE LANGUAGES (REL): Properties of recursive and recursively enumerable languages, Universal Turing machine, The Halting problem, Undecidable problems about TMs. Context sensitive language and linear bounded automata (LBA), Chomsky hierarchy, Decidability.

Teaching Methodology:

Teaching in this course is designed to engage the students in active and experimental learning by taking a problem solving and design oriented approach with special emphasis on real world applications. Students are expected to carry out lot of design and programming.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1, Unit-2

Test-2	25 Marks	Based on Unit-3 & Unit-4 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-5 to Unit-6 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Theory of Computation (will be added from time to time): Digital copy will be available on the JUET server.

Text Books:

1. K. L. P Mishra, N. Chandrashekaran (2003), Theory of Computer Science-Automata Languages and Computation, 2nd edition, Prentice Hall of India, India.
2. John E. Hopcroft, Rajeev Motwani, Jeffrey D. Ullman (2007), Introduction to Automata Theory Languages and Computation, 3rdedition, Pearson Education, India.

Reference Books:

1. Papadimitriou, Elements of the Theory of Computation, Prentice-Hall, 1998
2. Peter Dehning, Jack B. Dennis, “Machines, Languages and Computation”, Second Edition, Prentice-Hall, 1978
3. Harry R. Lewis, Christos H. Papadimitriou, "Elements of the theory of computation", Second Edition, Prentice-Hall, 1998

Title of Course: Minor Project-1 **Course Code: CS211**
Course Credits: 3

Course Learning Outcome:

After successful completion of this course student will be able to:

Course Outcome	Description
CO1	Acquire practical knowledge within the chosen area of technology for project development
CO2	Identify, analyze, formulate and handle programming projects with a comprehensive and systematic approach
CO3	Contribute as an individual or in a team in development of technical projects
CO4	Develop effective communication skills for presentation of project related activities

Syllabus:

A project to be developed based on one or more of the following concepts.

Introduction to Java 2 SDK Tool Set Object Oriented paradigm, arrays, collection objects, data types, variables, functions, Wrapper Class, Object Class Inheritance, Interfaces, Abstract Class, Inner Class Exception Handling, Customization of Exception classes Event Handling, Adapter Classes Introduction To Application Programming In Java2, Creating Window Application, Writing Console Application, Use of Utility and Math Packages Introduction To Swing, MVC Architecture, Swing AWT and JFC, Writing Swing Application, Swing Components, Changing Look and Feel of Application Enhancing Application Using Clipboard, Drag and Drop, I/O Stream Enhancement, Printing, Internationalization Garbage Collection and Application Cleanup Applet and Applet Security Network Programming, Sockets, URL Class, Internet Address Class Java database Programming, Java.Sql Package Study, JDBC, Different Types of Drivers of JDBC

Evaluation scheme:

Exam	Marks
P1	10 marks
P2	15 marks
P3	30 marks
Term paper	20 marks
Guide marks	25 marks (continuous evaluation-15, documentation-10)
Total	**100 marks**

Title of Course: Computer Networks Lab **Course Code: CS212**
L-T-P Scheme: 0-0-2 **Credits:** 1

Course Objectives

- To understand the working principle of various communication protocols.
- To analyze the various routing algorithms.
- To know the concept of data transfer between nodes.

Learning Outcomes:

Course Outcome	Description
CO1	Understand fundamental underlying principles of computer networking
CO2	Understand details and functionality of layered network architecture.
CO3	Apply mathematical foundations to solve computational problems in computer networking
CO4	Analyze performance of various communication protocols.
CO5	Compare routing algorithms
CO6	Practice packet /file transmission between nodes.

Course Content:

1. Identification of network hardware.
2. Fabrication of network cables and trouble shooting.
3. To study *stop &wait* and *sliding window* protocol.
4. To study MAC ALOHA protocol.
5. To study MAC CSMA and MAC CSMA/CD protocol.
6. To study TOKEN BUS and TOKEN RING.
7. To study ETHERNET.
8. To study TOKEN RING.
9. To study SWITCHED LAN.
10. To study Static routing.
11. To study dynamic routing.

Text Books:

1. Sybex CCNA Cisco Certified Network Associate Study Guide.5th Edition
2. Forouzen, Behrouz A.Fegan, Sophia Chung Data Communications and Networking TMH

Reference Books:

1. Carne, E. Bryan Professional's Guide to Data Communication in a CP/IP World Artech House, London, 2004

Title: Open Source Software Lab **Code: CS213**
L-T-P scheme: 0-0-2 **Credit: 1**

This Lab will be based on the subject run in CSE-Elective -1

Title of Course: Advanced Programming Lab-II **Course Code: CS214**

L-T-P scheme: 0-0-2 **Course Credit: 1**

Prerequisite: Students must have already registered for the course, "OOP".

Objective:

1. Demonstrate basic problem-solving skills: analyzing problems, modeling a problem as a system of objects, creating algorithms, and implementing models and algorithms in an object-oriented computer language (classes, objects, methods with parameters, abstract classes, interfaces, inheritance and polymorphism).
2. To learn using advanced features of a Programming Language.
3. To learn working with different APIs and make faster, reusable and efficient programs.

Learning Outcomes: At the end of the course students should:

1. Possess an ability to apply mathematical foundations, algorithmic principles, and computer science theory to the modeling and design of computer-based systems.
2. Be able to deconstruct problems to develop algorithms and eventually program code.
3. Develop substantial Java programs, when appropriate reusing previously created classes, writing programs requiring three or more classes.
4. Demonstrate ability to define the computing requirements of a problem and to design appropriate solutions based on established design principles and with an understanding of the tradeoffs involved in design choices.
5. Diligently leverage sound development principles to implement computer-based and software systems of varying complexity, and to evaluate such systems.

Course Contents:

Unit-1: Relooking classes, methods, objects, relationships, polymorphism, overriding and other object-oriented concepts.

Unit-2: Object oriented analysis and design, making inheritance-based designs, containership, abstract classes and interfaces.

Unit-3: Exception handling, polymorphic nature of exceptions, Multithreading, Race condition, Synchronization.

Unit-4: Introduction to design patterns, intents and class diagrams, singleton, factory, template, adapter pattern etc.

Unit-5: Project based on team work. The project may be menu-driven and should provide a design-oriented solution to a well-defined problem. The students should be able to identify the nature of the problem and perform object-oriented analysis followed by creating design solutions by identifying an appropriate design pattern. The code should justify the designs created.

References:

1. Horstmann, "CoreJava", Addison Wesley.
2. Urma, Fusco and Mycroft, "Java 8 In ACTION", Manning Publications, 1st edition, 2015.
3. Herbert Schieldt, "The Complete Reference: Java", TMH.
4. John Hunt, Alexander g. McManus,"Key Java: Advanced Tips and Techniques", Springer, 1998.
5. Y.Daniel Liang**, "**Introduction to Java programming", Comprehensive Version (9th Edition)
6. Cay S. Horstmann and Gary Cornell, "Core Java, Vol.2 Advanced Features" (8th Edition).

6th Semester

Title of Course: **Operating Systems** **Course Code: CS111**

L-T-P scheme: 3-1-0 **Credit: 4**

Prerequisite: Students must have knowledge of C programming and working of the computer systems.

Objective:

1. To familiarize with the basic functionality and the evolution of different types of operating systems.
2. To Learn and understand various algorithms related to CPU scheduling, deadlocks, memory management, and storage management.
3. To learn basic aspects of real time operating systems.

Learning Outcomes:

Course Outcome	Description
CO1	Gain knowledge of OS fundamentals along with process management concepts
CO2	Apply various process management concepts including scheduling, synchronization, dead-lock to solve given problem.
CO3	Explain various memory management techniques including virtual memory.
CO4	Analyse issues related to memory management.
CO5	Understand file system including disk structure by applying disk scheduling algorithm.
CO6	Work as a team on a project.

Course Content:

Unit-1: Introduction: Operating system structure, Operating system operations, Distributed systems, Special purpose systems, Computing environments, Open source operating systems.

Unit-2: CPU Scheduling: Process concepts: Process states, Process control block, Scheduling queues, Schedulers, Context switch, Multi-threaded programming: Overview, Multithreading models, Threading issues, Process scheduling: Basic concepts, Scheduling criteria, scheduling algorithms.

Unit-3: Synchronization: The Critical section problem, Synchronization hardware, Semaphores, Classic problems of synchronization, monitors.

Unit-4: Deadlocks: Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from deadlock.

Unit-5: Memory management: Memory management strategies, Swapping, Contiguous memory allocation, Paging, Structure of the page table, Segmentation.

Unit-6: Virtual Memory: Demand paging, copy on write, page replacement, allocation of frames, thrashing.

Unit-7: Storage Management: File concept, Access methods, directory structure, file system structure, directory implementation, allocation methods, free space management, disk structure, and disk-scheduling.

Unit-8: Case study on UNIX based Operating system: Design principles, Kernel modules, Process management, Memory management.

Unit-9: Real time systems: Characteristics of Real time operating systems, classification of real time systems, Micro kernels and RTOS, scheduling in RTOS, Rate monotonic scheduling, EDF, Priority inversion

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Based on Unit-1, Unit-2 & Unit-3
Test-2	25 Marks	Based on Unit-4 & Unit-5 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-6 to Unit-9 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Operating Systems (will be added from time to time): Digital copy will be available on the JUET server.

Text Book:

1. "Operating System Concepts"; A. Silberschatz , P. B. Galvin & G. Gagne , Wiley 10e 2018.
2. "Operating Systems: Internals and Design Principles"; W. Stallings, Pearson 9e, 2017.

Reference Books/Material:

1. "Real time systems design and analysis"; P. A. Laplante & S. J. Ovaska, Wiley, 2013.
2. "Real time systems: Theory and Practice"; Mall R., Pearson, 2e, 2009.

Title: Compiler Design **Code: 18B14CI541**

L-T-P scheme: 3-0-0 **Credit: 3**

Prerequisite:

Students must have already registered for the course, "Data Structures" and "Theory of Computation ".

Objective:

- Deepen the understanding of compiler design
- Develop problem solving ability using programming
- Develop ability to design and analyze a compiler

Learning Outcomes:

Course Outcome	Description
CO1	Specify and analyse the lexical, syntactic and semantic structures of advanced language features.
CO2	Separate the lexical, syntactic and semantic analysis into meaningful phases for a compiler to undertake language translation
CO3	Write a scanner, parser, and semantic analyser without the aid of automatic generators
CO4	Turn fully processed source code for a novel language into machine code for a novel computer
CO5	Describe techniques for intermediate code and machine code optimisation
CO6	Design the structures and support required for compiling advanced language features.

Course Content:

UNIT I: INTRODUCTION TO COMPILERS

Translators-Compilation and Interpretation, Language processors, The Phases of Compiler, Errors Encountered in Different Phases,The Grouping of Phases of Compiler , Programming Language basics.

UNIT II: LEXICAL ANALYSIS

Need and Role of Lexical Analyzer,Lexical Errors,Expressing Tokens by Regular Expressions, Converting Regular Expression to DFA ,Minimization of DFA, Language for Specifying Lexical Analyzers, LEX (Design of Lexical Analyzer for a sample Language).

UNIT III: SYNTAX ANALYSIS

Need and Role of the Parser, Context Free Grammars , Top Down Parsing ,General Strategies, Recursive Descent Parser ,Predictive Parser ,LL(1) Parser,Shift Reduce Parser-LR Parser, LR (0)Item, Construction of SLR Parsing Table , Introduction to LALR Parser ,Error Handling and Recovery in Syntax Analyzer, YACC (Design of a syntax Analyzer for a Sample Language) .

UNIT IV: SYNTAX DIRECTED TRANSLATION & RUN TIME ENVIRONMENT

Syntax directed Definitions, Construction of Syntax Tree, Bottom-up Evaluation of S-Attribute Definitions, Design of predictive translator, Type Systems, Specification of a simple type checker, Equivalence of Type Expressions, Type Conversions. RUN-TIME ENVIRONMENT: Source Language Issues, Storage Organization-Storage Allocation, Parameter Passing, Symbol Tables.

UNIT V: CODE OPTIMIZATION

Principal Sources of Optimization, DAG, Optimization of Basic Blocks, Global Data Flow Analysis, Efficient Data Flow Algorithms.

UNIT VI: CODE GENERATION

Issues in Design of a Code Generator, a Simple Code Generator Algorithm.

Teaching Methodology:

Teaching in this course is designed to engage the students in active and experimental learning by taking a problem solving and design oriented approach with special emphasis on real world applications. Students are expected to carry out lot of design and programming.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Based on Unit-1, Unit-2
Test-2	25 Marks	Based on Unit-3 & Unit-4 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-5 to Unit-6 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Compiler Design (will be added from time to time): Digital copy will be available on the JUET server.

Text Books:

1. Compilers : Principles, Techniques and Tools, Aho, Sethi and Ullman, Pearson Education
2. Principles Of Compiler Design by Alfred V Aho and Ullman, Narosa Publication

Reference Books:

1) Compiler Design in C, Holub, Prentice Hall of India

2) Advanced Compiler Design and Implementation, Muchnick Steven, Morgan Kauffman Publishers

3) Compiler Design, Santanu Chattopadhyay, PHI

4) Compiler Construction Principles and Practice, Kenneth C. Louden, Thomson

5) Compiler Construction and Design, Rajni Jindal , Umesh Publications

Title of Course: Minor Project-2 **Course Code: CS215**
Course Credits: 3

Course Learning Outcome:

After successful completion of this course student will be able to:

Course Outcome	Description
CO1	Analyze chosen literature addressing real world research problem to identify the requirements
CO2	Build technical report detailing the software specification, design, test plan, and implementation details.
CO3	Build a practicable solution for the research problem
CO4	Evaluate results to test the effectiveness of the proposed solution
CO5	Develop effective communication skills for presentation of project related activities

Syllabus:

A project to be developed based on one or more of the following concepts.

Project based learning: Each student in a group of 3-4 will have to develop a Minor Project based on different engineering concepts. The students can opt any real-world application for the implementation of Minor Project. The students have to implement the real world problem using any open-source programming language. Project development will enhance the knowledge and employability of the students in IT sector.

Evaluation scheme:

Exam	Marks
P1	10 marks
P2	15 marks
P3	30 marks
Term paper	20 marks
Guide marks	25 marks (continuous evaluation-15, documentation-10)
Total	**100 marks**

Title of Course: **Operating Systems Lab** **Course Code: CS216**

L-T-P scheme: 0-0-2 **Credit: 1**

Prerequisite: Students must have knowledge of C programming and working of the computer systems.

Objective:

1. To execute shell scripts in UNIX based operating system.
2. To implement inter process communication using system calls.
3. To implement algorithms for CPU scheduling as well as process synchronization learn and be able to implement the front-end and back-end web-technologies.

Learning Outcomes:

Course Outcome	**Description**
CO1	Understand and execute basic commands of shell script.
CO2	Apply basic operations in shell scripts which are required for different applications.
CO3	Identify and understand concept of file systems in shell script.
CO4	Apply concept of creating new process from parent process.
CO5	Apply concept of virtual file and execute basic commands on it.
CO6	Design communication mechanisms ipc and pipe on linux.

Course Content:

Unit-1; Comparative Study of different operating systems
Unit-2: Demonstration of multitasking concept.
Unit-3: Implementing various process creation algorithms(FCFS,SJF and Round-Robin Scheduling)
Unit-4: Implementation of memory allocation policies.
Unit-5: Implementing Page replacement algorithms (FIFO,LIFO)
Unit-6: Implementing segmentation algorithms
Unit-7: Implementing file-handling algorithms
Unit-8: Demonstration of working of distributed OS environment.

Evaluation Scheme:

<table>
<tr><th colspan="2">Exams</th><th>Marks</th><th>Coverage</th></tr>
<tr><td colspan="2">P-1</td><td>15 Marks</td><td>Based on Lab Exercises: 1-4</td></tr>
<tr><td colspan="2">P-2</td><td>15 Marks</td><td>Based on Lab Exercises: 5-8</td></tr>
<tr><td rowspan="4">Day-to-Day Work</td><td>Viva</td><td>20 Marks</td><td rowspan="4">70 Marks</td></tr>
<tr><td>Demonstration</td><td>20 Marks</td></tr>
<tr><td>Lab Record</td><td>15 Marks</td></tr>
<tr><td>Attendance & Discipline</td><td>15 Marks</td></tr>
<tr><td colspan="2">Total</td><td colspan="2">100 Marks</td></tr>
</table>

Learning Resources:

Study material of Web Technology Lab (will be added time to time): Digital copy will be available on the JUET server.

Text Book:

1. "Operating System Concepts"; A. Silberschatz , P. B. Galvin & G. Gagne , Wiley 10e 2018.
2. "Operating Systems: Internals and Design Principles"; W. Stallings, Pearson 9e, 2017.

Reference Books/Material:

1. "Real time systems design and analysis"; P. A. Laplante & S. J. Ovaska, Wiley, 2013.
2. "Real time systems: Theory and Practice"; Mall R., Pearson, 2e, 2009.

Title of Course: Advanced Programming Lab-3 **Course Code: CS217**

L-T-P scheme:0-0-2 **Credit:1**

Prerequisite:

- Students should have a strong understanding of at least one programming language (e.g., Java, Python, C++).
- Proficiency in fundamental data structures (arrays, linked lists, stacks, queues) and algorithms is essential,
- Understanding of OOP concepts like classes, objects, inheritance, polymorphism, and encapsulation.
- Familiarity with basic software development concepts, including version control (e.g., Git), debugging, and testing.
- Basic knowledge of databases and SQL
- Familiarity with web development concepts (HTML, CSS, basic JavaScript) can be beneficial.

Objectives:

1. To empower students with advanced programming skills by immersing them in the principles of software design. By the end of the course, students will demonstrate the ability to apply these concepts in real-world scenarios, resulting in the creation of well-designed, modular, and maintainable software solutions.
2. To equip students with a comprehensive understanding and practical proficiency in modern software development concepts, focusing on Docker and microservices, enabling them to design, implement, and deploy scalable and maintainable software solutions.

Learning Outcomes:

Course Outcome	Description
CO1	Students will demonstrate the ability to follow systematic software design steps as instructed by the instructor. This includes understanding and adhering to a structured approach in the software development process.
CO2	Students will apply software design concepts, including principles of abstraction, modularity, and encapsulation, in practical scenarios. They will showcase the ability to translate theoretical knowledge into practical application.
CO3	Students will demonstrate adaptability to emerging technologies in the software development landscape, showcasing competence in the use of modern tools and methodologies
CO4	Students will demonstrate strong problem-solving skills and critical thinking, applying these skills to propose effective solutions for complex engineering problems.
CO5	Students will adhere to ethical considerations and demonstrate their communication skills in their coding practices, project presentations, and collaborative work.

Course Content:

Unit-1: Understanding Software Design, Definition of software design, Importance of good design in software development, Basic Principles of Software Design, Abstraction, Modularity, Encapsulation, Separation of Concerns (SoC)

Unit-2: Cohesion Types of cohesion (functional, sequential, communicational, procedural, temporal, logical), Strategies to achieve high cohesion, Coupling, Types of coupling (data coupling, control coupling, stamp coupling, data-structure coupling, control-structure coupling), Strategies to achieve loose coupling, SOLID design principles

Unit-3: Introduction to Design Patterns, Benefits of using design patterns, Common Design Patterns, Creational Patterns (Singleton, Factory, Abstract Factory), Structural Patterns (Adapter, Decorator, Proxy), Behavioral Patterns (Observer, Strategy, Command)

Unit-4: Introduction to Docker, Benefits of containerization, working with Docker, Docker containers, Docker images, Docker Compose

Unit-5: Microservices Overview, Characteristics and advantages of microservices architecture, Building Microservices, Design considerations, Communication between microservices, Deployment strategies for microservices, Docker and Microservices, Containerizing microservices, Orchestration with Docker Swarm or Kubernetes,

Evaluation Scheme:

<table>
<tr><th colspan="2">Exams</th><th>Marks</th><th>Coverage</th></tr>
<tr><td colspan="2">P-1</td><td>15 Marks</td><td>Uni1-1 & Unit-2</td></tr>
<tr><td colspan="2">P-2</td><td>15 Marks</td><td>Unit-3, Unit-4 & Unit-5</td></tr>
<tr><td rowspan="4">Day-to-Day Work</td><td>Viva</td><td>20 Marks</td><td rowspan="4">70 Marks</td></tr>
<tr><td>Demonstration</td><td>20 Marks</td></tr>
<tr><td>Lab Record</td><td>15 Marks</td></tr>
<tr><td>Attendance & Discipline</td><td>15 Marks</td></tr>
<tr><td colspan="2">Total</td><td colspan="2">100 Marks</td></tr>
</table>

Resources:

- "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin
- "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
- Docker documentation: https://docs.docker.com/
- Microservices architecture: Various online resources and case studies.

7th Semester:

Title: Major Project Part – I **Code: CS218**

L-T-P scheme: 0**-0-8** **Credit: 04**

Prerequisite: Students must have knowledge about software development.

Objective:

1. To apply engineering knowledge in practical problem solving
2. To foster innovation in design of products, processes or systems
3. To develop creative thinking in finding viable solutions to engineering problems.

Learning Outcomes:

Course Outcome	Description
CO1	Summarize the contemporary literature and explore tools for hands-on in the respective project area
CO2	List out the specific requirements to develop the workable solution for the identified computing problem.
CO3	Develop a working model for the identified problem
CO4	Inspect the developed solution using exhaustive test cases and evaluate its performance using statistical methods and relevant metrics
CO5	Compile the results and findings of the project in written and verbal formats
CO6	Report the results and findings of the project in written and verbal formats.

Course Content:

Project based learning: Each student in a group of 2-3 will have to develop a Major Project based on different real-world problems using any open-source programming language. Students have to study the state-of-the-art methods before finalizing the objectives. Project development will enhance the knowledge and employability of the students in IT sector.

Teaching Methodology:

- Regular supervision by project guide

Evaluation Scheme:

Exams	Marks	Coverage
Mid Semester Viva	20 Marks	Based on Unit-1, Unit-2, Unit-3
Final Viva	30 Marks	Based on Unit-4,Unit-5,Unit-6 and Unit-7
Project Report	20 Marks	
Day to Day Work	30 Marks	
Total	**100 Marks**	

Text Book/Reference material:

- Seven latest international journal papers having high impact factor. MOOC Courses from Coursera, NPTEL etc.

Journals References:

[1] ieee.org
[2] dl.acm.org
[3] Elsevier
[4] Springer

8th Semester:

Title: Major Project Part – II **Code: CS219**

L-T-P scheme: 0**-0-16** **Credit: 08**

Prerequisite: Students must have already completed Project Part-1.

Objective:

1. To apply engineering knowledge in practical problem solving
2. To foster innovation in design of products, processes or systems
3. To develop creative thinking in finding viable solutions to engineering problems.

Learning Outcomes:

Course Outcome	Description
CO1	Summarize the contemporary scholarly literature, activities, and explored tools for hands-on in the respective project area
CO2	List out the specific requirements to develop the workable solution for the identified computing problem.
CO3	Develop a workable computing solutions for the identified problem
CO4	Evaluate the performance of the developed solution
CO5	Compile the results and findings of the project in written and verbal formats
CO6	Developing the ability of develop a complete IR system from scratch.

Course Content:

Unit-1: In depth study of the topic assigned in the light of the preliminary report prepared in the seventh semester.

Unit-2: Review and finalization of the approach to the problem relating to the assigned topic.

Unit-3: Preparing a detailed action plan for conducting the investigation, including team work Detailed Analysis/Modeling/Simulation/Design/Problem Solving/Experiment as needed.

Unit-4: Final development of product/process, testing, results, conclusions and future directions.

Unit-5: Preparing a paper for Conference presentation/Publication in Journals, if possible.

Unit-6: Preparing a report in the standard format for being evaluated by the dept. assessment board.

Unit-7: Final project presentation and viva voce by the assessment board including external expert.

Teaching Methodology:

- Regular supervision by project guide

Evaluation Scheme:

Exams	Marks	Coverage
Mid Semester Viva	20 Marks	Based on Unit-1, Unit-2, Unit-3
Final Viva	30 Marks	Based on Unit-4,Unit-5,Unit-6 and Unit-7
Project Report	20 Marks	
Day to Day Work	30 Marks	
Total	**100 Marks**	

Text Book/Reference material:

- Seven latest international journal papers having high impact factor. MOOC Courses from Coursera, NPTEL etc.

Journals References:

[1] ieee.org
[2] dl.acm.org
[3] Elsevier
[4] Springer

Course Description

CSE Electives:

Course Name: Mathematics for Artificial Intelligence **Course Code:** CS301
L-T-P scheme: 3-0-0 **Credits:** 3

Prerequisite:
Students should be familiar with basic linear algebra, calculus, and probability theory.

Objective:

This course is aimed:

- To develop mathematical maturity essential for understanding machine learning and AI algorithms.
- To introduce students to the core concepts of linear algebra, probability, optimization, and statistics relevant to AI.
- To prepare students for advanced studies and research in Artificial Intelligence.

Learning Outcomes:

Course Outcome	Description
CO1	Understand the fundamentals of linear algebra including vector spaces, eigenvalues, and singular value decomposition.
CO2	Apply concepts of probability, random variables, and distributions to model uncertainty in AI.
CO3	Grasp the basics of optimization including gradient descent, convexity, and constraints handling.
CO4	Use calculus in understanding neural networks and backpropagation.
CO5	Model and analyze data using statistics, expectation, variance, and hypothesis testing.
CO6	Apply mathematical tools to practical problems in machine learning and artificial intelligence.

Course Contents:

Unit 1: Linear Algebra for AI
Vectors, matrices, operations, vector spaces, rank, linear independence, eigenvalues and eigenvectors, SVD, applications in data representation and PCA.

Unit 2: Probability and Random Variables
Probability theory basics, conditional probability, Bayes theorem, random variables, probability distributions, expectation, variance, covariance.

Unit 3: Calculus and Optimization
Functions, limits, derivatives, partial derivatives, gradients, chain rule, Taylor series, convexity, gradient descent, constrained optimization.

Unit 4: Statistics for AI
Descriptive statistics, statistical inference, hypothesis testing, confidence intervals, p-values, correlation and regression analysis.

Unit 5: Applications in Machine Learning
Using mathematical tools in supervised and unsupervised learning, linear regression, logistic regression, support vector machines, neural networks basics.

Methodology:

The course will be taught through lectures integrated with problem-solving sessions. Students will be provided with tutorials and assignments. Applications in real-world machine learning problems will be emphasized through examples and case studies.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered up to Test-1
Test-2	25 Marks	Syllabus covered up to Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Books:

1. Gilbert Strang, *Linear Algebra and Its Applications*, Cengage Learning.
2. Sheldon Ross, *A First Course in Probability*, Pearson.
3. Trevor Hastie, Robert Tibshirani, Jerome Friedman, *The Elements of Statistical Learning*, Springer.
4. David C. Lay, *Linear Algebra and Its Applications*, Pearson.
5. Marc Peter Deisenroth, A. Aldo Faisal, Cheng Soon Ong, *Mathematics for Machine Learning*, Cambridge University Press.

Course Name: Big Data Systems **Course Code:** CS302
L-T-P scheme: 3-0-0 **Credits:** 3

Prerequisite:
Students should have knowledge of database systems, computer networks, and basic programming in Java/Python.

Objective:

This course is aimed:

- To introduce the architecture, challenges, and technologies in Big Data systems.
- To familiarize students with distributed computing paradigms including MapReduce and Spark.
- To develop the ability to work with large datasets and apply scalable algorithms.
- To provide practical skills in handling, processing, and analyzing big data.

Learning Outcomes:

Course Outcome	Description
CO1	Understand the characteristics, architecture, and challenges of Big Data systems.
CO2	Work with distributed file systems (e.g., HDFS) and understand data storage mechanisms for large-scale systems.
CO3	Develop and optimize programs using MapReduce and Apache Spark.
CO4	Design and manage NoSQL databases for big data applications.
CO5	Apply big data processing techniques in data analytics and real-time processing scenarios.
CO6	Evaluate and choose appropriate big data tools for specific application needs.

Course Contents:

Unit 1: Introduction to Big Data
Big Data characteristics: volume, velocity, variety, veracity, and value. Big Data architecture and components. Data sources and data types. Challenges in Big Data processing.

Unit 2: Hadoop Ecosystem and HDFS
Hadoop architecture, HDFS structure and operations. Fault tolerance and replication. Hadoop ecosystem components: YARN, Hive, Pig, Sqoop, Flume.

Unit 3: MapReduce Programming
MapReduce programming model, examples, and performance tuning. Combiners and partitioners. Applications of MapReduce in real-world problems.

Unit 4: Apache Spark
Introduction to Spark, RDDs, Spark SQL, and DataFrames. Transformations and actions. Spark Streaming. Machine Learning with MLlib.

Unit 5: NoSQL and Big Data Storage
Types of NoSQL databases: key-value, column-based, document-oriented, graph databases. Introduction to Cassandra, MongoDB, and HBase. Data modeling and indexing in NoSQL.

Unit 6: Big Data Applications and Tools
Case studies from e-commerce, social networks, IoT. Real-time processing tools: Apache Kafka, Storm. Big Data security and privacy concerns.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered up to Test-1
Test-2	25 Marks	Syllabus covered up to Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Books:

1. Tom White, *Hadoop: The Definitive Guide*, O'Reilly Media.
2. Holden Karau, Andy Konwinski, Patrick Wendell, *Learning Spark*, O'Reilly Media.
3. Alex Holmes, *Hadoop in Practice*, Manning Publications.
4. Chuck Lam, *Hadoop in Action*, Manning Publications.
5. Pramod J. Sadalage and Martin Fowler, *NoSQL Distilled*, Addison-Wesley.

Supplementary materials including slides, datasets, and tutorials will be provided on the university server.

Title of Course: Artificial Intelligence **Course Code: CS303**
L-T Scheme: 3-1 **Course Credits: 4**

Objectives: In this course we will study the basic components of an intelligent system, their functions, mechanisms, policies and techniques used in their implementation and examples.

Learning Outcomes: The students will have a detailed knowledge of the concepts of artificial intelligence, various applications of AI in different fields, Aware of a variety of approaches to AI techniques.

Course Outcome	Description
CO1	Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents.
CO2	Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game based techniques to solve them.
CO3	Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing.
CO4	Attain the capability to represent various real life problem domains using logic based techniques and use this to perform inference or planning. playing.
CO5	Formulate and solve problems with uncertain information using Bayesian approaches.
CO6	Apply concept Natural Language processing to problems leading to understanding of cognitive computing.

Course Contents:

Unit-1 (Introduction to AI):
Definitions, Goals of AI, AI Approaches, AI Techniques, Branches of AI, Applications of AI. Introduction of Intelligent Systems: Agents and Environments, Good Behavior: the concept of Rationality, The Nature of Environments, The structure of Agents, How the components of agent programs work.

Unit-2 (Problems Solving, Search and Control Strategies)
Solving Problems by Searching, Study and analysis of various searching algorithms. Implementation of Depth-first search, Problem-Solving Agents, Searching for Solutions, Uninformed Search Strategies: Breadth-first search, Uniform-cost search, Depth-first search, Depth-limited search, Iterative deepening depth-first search, Bi-directional search Informed (Heuristic) Search Strategies: Greedy best-first search A* search: Minimizing the total estimated solution cost, Conditions for optimality: Admissibility and consistency, Optimality of A*, Memory-bounded heuristic search, Heuristic Functions, Generating admissible heuristics from sub problems: Pattern databases, Learning heuristics from experience.

Beyond Classical Search: Local Search Algorithms and Optimization Problems: Hill-climbing search Simulated annealing, Local beam search, Genetic algorithms, Local Search in Continuous Spaces, Searching with Non-deterministic Actions: AND-OR search trees, Searching with Partial Observations.

Adversarial Search and Constraint Satisfaction Problems, Study of min-max algorithm Adversarial Search: Games, Optimal Decisions in Games, The mini-max algorithm, Optimal decisions in multiplayer games, Alpha--Beta Pruning, Move ordering , Imperfect Real-Time Decisions, Evaluation functions, Cutting off search, Forward pruning, Search versus lookup, Stochastic Games, Evaluation functions for games of chance, Partially Observable Games

Constraint Satisfaction Problems: Defining Constraint Satisfaction Problems, Variations on the CSP formalism, Constraint Propagation: Inference in CSPs, Backtracking Search for CSPs, Local Search for CSPs, Alpha-beta pruning and CSP, Implementation aspects of mini-max algorithm and CSP.

Unit- 3 (Knowledge Representations Issues, Predicate Logic, Rules)
Knowledge representation, KR using predicate logic, KR using rules. Reasoning System - Symbolic, Statistical: Reasoning, Symbolic reasoning, Statistical reasoning.

Unit-4 (Quantifying Uncertainty, Learning Systems)
Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Bayes' Rule and Its Use, Representing Knowledge in an Uncertain Domain, Other Approaches to Uncertain Reasoning, Rule-based methods for uncertain reasoning, Representing vagueness: Fuzzy sets and fuzzy logic, Study of fuzzy logic and Decision trees, Implementation aspects of Decision trees.

Learning from Examples: Forms of Learning, Supervised Learning, Learning Decision Trees, The decision tree representation, Expressiveness of decision trees, inducing decision trees from examples.

Unit-5 (Expert Systems)
Introduction, Knowledge acquisition, Knowledge base, Working memory, Inference engine, Expert system shells, Explanation, Application of expert systems.

Fundamentals of Neural Networks: Introduction and research history, Model of artificial neuron, Characteristics of neural networks, learning methods in neural networks, Single-layer neural network system, Applications of neural networks.
Fundamentals of Genetic Algorithms: Introduction, Encoding, Operators of genetic algorithm, Basic genetic algorithm.

Evalution Scheme:

Evaluation	**Marks**	**Remarks**
T1	15 Marks(1 Hr.)	1^{st} -4^{th} Week
T2	25 Marks(1:30 Hr.)	5^{th} – 10^{th} Week
T3	35 Marks(2:00 Hr.)	11^{th} -16^{th} Week
Tutorial/Presentation	10	
Assignments	5	
Quiz	5	
Attendance	5	
Total	**100**	

Text Books

1. Rich, Elaine Knight, Kevin, Artificial Intelligence, Tata McGraw Hill.
2. Luger, George F, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education.

References

1. Nilsson, Nils J, Artificial Intelligence, Morgan Kaufmann, Russell, Stuart J. Norvig, Peter, AI: A Modern Approach, Pearson Education.
2. Neural networks and Learning Machines, Simon Haykin, PHI Learning Pvt. Ltd.

Course Name: Data Analysis and Visualization **Course Code:** CS304
L-T-P scheme: 3-0-0 **Credits:** 3

Objective:

This course is aimed:

- Explain the concept of visualization in the processing and analysis of data.
- Develop visualization methods and visualization systems using software applications.
- Perform creative work in the field of visualization.

Learning Outcomes:

Course Outcome	Description
CO1	Manipulate large datasets and handle missing or inconsistent values in datasets.
CO2	Perform statistical analysis using numpy and scipy.
CO3	Discover and visualize datasets using seaborn and matplotlib.
CO4	Perform machine learning using Scikit-learn.

Course Contents:

Unit 1: Introduction to visualization
Introduction of visual perception, Visual representation of data, Data Abstraction, Visual Encodings, Use of Color, Perceptual Issues, Information overloads.

Unit 2: Creating visual representations
Visualization reference model, Visual mapping, Visual analytics, Design of Visualization applications.

Unit 3: Non spatial data visualization
Visualization of one, two and multi-dimensional data, Tabular data, quantitative values (scatter plot), Separate, Order, and Align (Bar, staked Bar, dots and line charts), Tree data, Displaying Hierarchical Structures, graph data, rules for graph drawing and labeling, text and document data, levels of text representation, visualizations of a single text document, word cloud, flow data. Time series data, characteristics of time data, visualization time series data, mapping of time

Unit 4: Spatial Data Visualization
Scalar fields, Isocontours (Topographic Terrain Maps), scalar volumes, Direct Volume Rendering (Multidimensional Transfer Functions) , Maps(dot, pixel), vector fields Defining Marks and Channels

Unit 5: Software tools and data for visualization
The iris data set, The Detroit Data Set, The Breakfast Cereal Data Set, The Dow Jones Industrial Average Data set (time series), MS Spreadsheet, Python, Matlab, Java, Tableau

Methodology:
The course will be taught through lectures integrated with problem-solving sessions. Students will be provided with tutorials and assignments. Applications in real-world problems will be emphasized through examples and case studies.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Syllabus covered up to Test-1
Test-2	25 Marks	Syllabus covered up to Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Books:

1. Fry, Visualizing Data. O'Reilly Media, 2008, ISBN 0596514557
2. Ware, Information Visualization: Perception for Design, 3rd ed. Morgan Kaufmann, 2012
3. Telea, Data Visualization: Principles and Practice. A. K. Peters, Ltd, 2007, ISBN 1568813066.

Title of Course: Neural Network **Course Code: CS305**
L-T Scheme: 3-0-0 **Course Credits: 3**

Objectives: To introduce some of the fundamental techniques and principles of neural computation and to investigate some common models and their applications.

Prerequisites:
Basic knowledge of computer architecture, basics of algorithms.

Learning Outcomes:
On completion of this course, a student should be able to:

Course Outcome	Description
CO1	Understand the learning and generalization issue in neural computation.
CO2	Understand the basic ideas behind most common learning algorithms for multilayer perceptrons, radial-basis function networks, and Kohonen self-organising maps.
CO3	Implement common learning algorithms using an existing package
CO4	Apply neural networks to classification and recognition problems

Course Content:

Unit 1: What Are Neural Networks:
History, Artificial and biological neural networks, Artificial intelligence and neural networks.

Unit 2: Neurons and Neural Networks
Biological neurons, Models of single neurons, Different neural network models.

Unit 3: Single Layer Perceptrons
Least mean square algorithm, Learning curves, Learning rates, Perceptron

Unit 4: Multilayer Perceptrons
The XOR problem, Back-propagation algorithm, Heuristic for improving the back-propagation algorithm, Some examples

Unit 5: Radial-Basis Function Networks
Interpolation, Regularization, Learning strategies

Unit 6: Kohonen Self-Organizing Maps
Self-organizing map, The SOM algorithm, Learning vector quantization

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Syllabus covered upto T-1
Test-2	25 Marks	Syllabus covered upto T-2
Test-3	35 Marks	Entire Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Text Books

1. Introduction to Artificial Neural Systems, by Jacek Zurada
2. An Introduction to Neural Networks K. Gurney, UCL Press, London.
3. Introduction to Neural Networks, R. Beale and T. Jackson, IOP Press.
4. The Essence of Neural Networks, R. Callan, Prentice Hall Europe.
5. Neural Networks: A Comprehensive Foundation, Simon Haykin, Prentice Hall.
6. Book by Haykins
7. Book by Hassoul
8. Book by Yagnanarayana
9. Perceptrons, by Minsky and Papert
10. Parallel and Distributed Processing, by McClelland and Rumelhart
11. Neuro Computing - Volume 1 and Volume 2, edited by Anderson

Journals

1. IEEE transactions on Neural Networks
2. IEEE transactions on Systems, Man and Cybernetics (SMC)
3. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)
4. Neural Networks
5. Neuro Computing
6. Machine Learning

Title of Course: Cryptography **Course Code:** CS306
L-T Scheme: 3-0-0 **Course Credits:** 3

Objectives:
The course introduces students to the principles of cryptography, various cryptographic algorithms, their applications in distributed systems, and performance evaluation of cryptographic methods. It emphasizes both theoretical underpinnings and practical implementations in real-world scenarios.

Learning Outcomes:
Upon successful completion of this course, students will be able to identify and categorize cryptographic algorithms, analyze their suitability for various applications, implement cryptographic protocols, and evaluate their performance using established metrics and cryptanalysis methods.

Course Outcome	Description
CO1	Define the principle of cryptography along with the categorization of cryptographic algorithms and its applicability into various allied areas.
CO2	Understand the various cryptographic problems in distributed applications and its solutions such as cryptography, hashing, and digital signatures.
CO3	Verify the feasibility and applicability of different cryptography and security algorithms in distributed applications.
CO4	Perform the various cryptoanalysis algorithms like ElGamal, ECC, etc., for various distributed applications.
CO5	Evaluate the performance for various applications using various cryptographical algorithms and other related secure technologies.

Course Contents:

Unit 1: Introduction to Cryptography
Cryptography in the modern era, history of ciphers along with their cryptanalysis, rigorous versus heuristic approaches; principles of defining security and its adversarial models, Perfect Secrecy and its limitations.

Unit 2: Categorization of Cryptographic Algorithms
Categories of cryptographic algorithms, conceptual security, introduction to public and private key cryptography and its applications.

Unit 3: Symmetric Cryptography Models
Computational securities, definition of secure encryption, how to construct secure encryption, pseudo randomness, construction of CPA-secure encryption, illustration of CCA attacks.

Unit 4: Message Authentication
Differentiate between secrecy and integrity, pseudorandom generators, DES, AES, hash and MAC functions, RC4, CBC-MAC, HMAC, password hashing.

Unit 5: Number Theory and Asymmetric Key Cryptography
Fundamentals of group theory, factorization, primes and RSA, cryptographic assumptions in cyclic groups, hash functions to collision resistance with discrete log, introduction to public key encryption, Diffie-Hellman key exchange.

Unit 6: Public Key Encryption
Public key encryption systems and its definitions, hybrid model of encryption and KEM/DEM, El Gamal encryption, RSA: textbook encryption, attacks on textbook RSA, padded RSA; CCA-secure RSA KEM.

Unit 7: Elliptic Curve Cryptography (ECC) and Cryptoanalysis
Elliptic curve over finite fields, elliptic curve cryptosystems (Diffie-Hellman, ElGamal), elliptic curve digital signatures (ECDSA, Bitcoin), elliptic curve factorization, pairing-based systems and review.

Unit 8: Analysis of Various Cryptographic Signatures
Digital signature definition and its applications, RSA signatures: textbook RSA, hashed RSA, security with ROM, digital certificates, certificates and public-key infrastructures, proxy signature, Kerberos.

Unit 9: Cryptographic Evaluation Techniques
Constructions of pseudorandom permutations (block ciphers) in practice, substitution-permutation and Feistel networks, DES and attacks on reduced-round versions, double-DES and triple-DES, security of CTR with n - k bit counter for messages to size 2k blocks with proof directly to the LR definition, CCA attacks, birthday attacks, the random oracle model.

Project-Based Learning:
Projects will be based on secure communication protocols, cryptographic tool development, and performance evaluation of algorithms under various threat models.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Syllabus covered up to Test-1
Test-2	25 Marks	Syllabus covered up to Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Text Books:

1. D. R. Stinson, Paterson M., Cryptography: Theory and Practice, CRC Press, 2018

2. Keith Martin, Everyday Cryptography: Fundamental Principles and Applications, Oxford University Press, 2017

References:

1. Cryptography: Portable technology offers boost for nuclear security, arms control applications

2. Journal of Cryptography

3. ACM Transactions on Information and System Security

4. IEEE Press Computer Security and Privacy

5. IEEE Transactions on Information Forensics and Security

Course Name: Blockchain Technology **Course Code:** CS307
L-T-P scheme: 3-0-0 **Credits:** 3

Prerequisite:
Students should have basic understanding of computer networks, cryptography, and programming fundamentals.

Objective:

This course is aimed:

- To introduce the foundational concepts of blockchain and distributed ledger technology.
- To understand the underlying cryptographic principles and consensus mechanisms.
- To explore blockchain platforms like Bitcoin and Ethereum, and smart contracts.
- To provide insight into real-world blockchain applications and challenges.

Learning Outcomes:

Course Outcome	Description
CO1	Understand the architecture, design principles, and evolution of blockchain technology.
CO2	Explain the use of cryptographic tools such as hashing and digital signatures in blockchain.
CO3	Analyze consensus algorithms and their impact on the security and performance of blockchain systems.
CO4	Develop and deploy smart contracts using platforms like Ethereum and Solidity.
CO5	Explore real-world use cases and applications of blockchain in areas like finance, supply chain, and identity management.
CO6	Evaluate scalability, security, and privacy challenges in blockchain systems.

Course Contents:

Unit 1: Introduction to Blockchain
History and evolution. Centralized vs. decentralized systems. Basic architecture and components. Distributed ledger, blocks, nodes, miners.

Unit 2: Cryptography in Blockchain
Hash functions (SHA-256), Merkle trees, public key cryptography, digital signatures, proof of work.

Unit 3: Consensus Mechanisms
Distributed consensus and its challenges. Proof of Work (PoW), Proof of Stake (PoS), Delegated PoS, Practical Byzantine Fault Tolerance (PBFT), mining and block propagation.

Unit 4: Bitcoin and Ethereum
Bitcoin protocol, transactions, blocks, wallets, mining economics. Ethereum architecture, Ethereum Virtual Machine (EVM), Gas, smart contracts.

Unit 5: Smart Contracts and Solidity
Solidity programming basics, developing and deploying smart contracts, event handling, testing with Remix and Truffle.

Unit 6: Applications and Challenges
Applications in finance (DeFi), healthcare, identity, supply chain. Blockchain in IoT and voting. Security and privacy issues. Limitations and future directions (Layer-2 solutions, interoperability, scalability).

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered up to Test-1
Test-2	25 Marks	Syllabus covered up to Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Books:

1. Melanie Swan, *Blockchain: Blueprint for a New Economy*, O'Reilly.
2. Andreas M. Antonopoulos, *Mastering Bitcoin*, O'Reilly Media.
3. Andreas M. Antonopoulos & Gavin Wood, *Mastering Ethereum*, O'Reilly Media.
4. Imran Bashir, *Mastering Blockchain*, Packt Publishing.
5. Joseph Bonneau et al., *SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies*, IEEE Security & Privacy.

Course Name: Intrusion Detection and Prevention Systems **Course Code:** CS308
L-T-P scheme: 3-0-0 **Credits:** 3

Prerequisite:
Students should have basic knowledge of computer networks, operating systems, and cybersecurity fundamentals.

Objective:

This course is aimed:

- To provide foundational knowledge of intrusion detection and prevention systems (IDPS).
- To understand attack vectors, detection techniques, and prevention strategies.
- To explore architecture and components of host-based and network-based IDPS.
- To analyze modern IDPS tools and their deployment in enterprise environments.

Learning Outcomes:

Course Outcome	Description
CO1	Understand the need, objectives, and classification of intrusion detection and prevention systems.
CO2	Analyze network and host-based threats, and understand how attacks are detected using signature-based and anomaly-based methods.
CO3	Design and evaluate different types of IDPS architectures and deployment strategies.
CO4	Apply data mining and machine learning techniques for intrusion detection.
CO5	Work with tools such as Snort, Suricata, Bro/Zeek for monitoring and alerting suspicious activities.
CO6	Explore challenges in IDPS such as false positives, performance, and evasion techniques.

Course Contents:

Unit 1: Introduction to IDPS
Basic concepts, need for intrusion detection, taxonomy of IDPS (NIDS, HIDS, hybrid), architecture, and components. Comparison between detection and prevention systems.

Unit 2: Types of Intrusions and Attacks
Classification of intrusions: malware, DoS, scanning, insider threats. Attack signatures. Footprinting and reconnaissance techniques. Threat modeling.

Unit 3: Detection Techniques
Signature-based detection, anomaly-based detection, heuristic and stateful protocol analysis. Detection metrics: true/false positives, ROC curves. Machine learning approaches in IDS.

Unit 4: IDPS Architecture and Deployment
Centralized vs. distributed architecture, sensor placement, inline vs. passive monitoring, IDS lifecycle, response mechanisms, integration with SIEM tools.

Unit 5: IDPS Toolsand Technologies
Overview and configuration of Snort, Suricata, Zeek/Bro. Log analysis and rule writing. Traffic monitoring with Wireshark. Correlation and alert generation.

Unit 6: Challenges and Trends in IDPS
Limitations of IDPS, evasion techniques, encrypted traffic inspection, scalability. Emerging trends: AI-based IDPS, cloud-native intrusion detection, threat intelligence integration.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered up to Test-1
Test-2	25 Marks	Syllabus covered up to Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Books:

1. Rebecca Gurley Bace, *Intrusion Detection*, Macmillan Technical Publishing.
2. Rafeeq Rehman, *Intrusion Detection Systems with Snort*, Prentice Hall.
3. Paxson, Vern, *Bro: A System for Detecting Network Intruders in Real-Time*, USENIX.
4. Northcutt & Novak, *Network Intrusion Detection*, New Riders.
5. Chris Sanders, *Applied Network Security Monitoring*, Elsevier.

Course Name: Software Testing **Course Code:** CS309
L-T-P scheme: 3-0-0 **Credits:** 3

Prerequisite:
Students should have prior knowledge of software engineering principles, programming, and data structures.

Objective:

This course is aimed:

- To introduce systematic approaches for software testing in different phases of development.
- To understand test planning, test case design, test execution, and result analysis.
- To explore automated testing, debugging, and quality assurance techniques.
- To equip students with practical tools and methods to ensure software reliability and performance.

Learning Outcomes:

Course Outcome	Description
CO1	Understand the principles, scope, and objectives of software testing, and its integration into the software development process.
CO2	Design efficient and effective test cases using structured black-box and white-box techniques.
CO3	Analyze and apply advanced testing strategies at unit, integration, and system levels.
CO4	Conduct performance, security, and usability testing to evaluate software quality in real-world scenarios.
CO5	Plan and manage test activities, track defects, and apply quality assurance metrics and industry standards.

Course Contents:

Unit 1: Foundations of Software Testing
Definition, goals, and taxonomy of testing. Software Testing Life Cycle (STLC). Roles and responsibilities in testing teams. Verification vs. Validation. Manual vs. automated testing. Testing in different SDLC models (Waterfall, Agile, V-Model). Defect classification and reporting.

Unit 2: Test Case Design and Static Techniques
Black-box test design: Equivalence class partitioning, Boundary value analysis, Cause-effect graphing, State transition testing. White box test design: Control flow testing, Decision / Condition coverage, Loop testing, Cyclomatic complexity. Static testing techniques: Code reviews, walkthroughs, inspections, formal technical reviews.

Unit 3 :Levels and Types of Testing
Unit Testing – techniques and frameworks. Integration testing – interface testing, stubs and drivers. System testing – functional and non-functional validation. Acceptance testing – alpha, beta, user acceptance testing. Regression testing strategies. Exploratory and ad-hoc testing.

Unit 4: Performance, Security & Usability Testing
Performance testing – load, stress, volume, spike testing. Monitoring and analysis using tools like Apache JMeter. Security testing – threat modeling, vulnerability scanning, penetration testing basics.Tools: OWASP ZAP.Usability and accessibility testing – user experience testing, UI standards, WCAG guidelines. Case studies from enterprise applications.

Unit 5: Test Management and Quality Metrics
Test planning and estimation. Risk-based testing. Requirement traceability matrix. Test metrics – defect density, test effectiveness, test coverage. Defect life cycle and defect tracking systems. Overview of IEEE 829 documentation standard. Introduction to quality models – ISO 9126, Six Sigma in testing.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered up to Test-1
Test-2	25 Marks	Syllabus covered up to Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Books:

1. Paul C. Jorgensen, *Software Testing: A Craftsman's Approach*, CRC Press.
2. Ron Patton, *Software Testing*, Pearson Education.
3. Glenford J. Myers, *The Art of Software Testing*, Wiley.
4. Srinivasan Desikan & Gopalaswamy Ramesh, *Software Testing: Principles and Practices*, Pearson.
5. Aditya P. Mathur, *Foundations of Software Testing*, Pearson.

Title of Course: Web Engineering **Course Code: CS310**
L-T-P Scheme: 3-0-0 **Course Credits: 3**

Prerequisite: Students must have already registered for the courses, Software Engineering, Web Technology Lab.

Objective: To develop an ability to design and implement static and dynamic web-applications and mobile applications.

Web Engineering	
Course Outcome	**Description**
CO1	Outline various terminologies of web development based engineering approaches.
CO2	Describe the real world problems and able to identify suitable solution in terms of appropriate web development models.
CO3	Understanding the customer requirements and the complexities that may arise in achieving these requirements in web development.
CO4	Develop and analyze the approaches for designing web based applications
CO5	Identify and use various tools in various processes in web based application development
CO6	Apply suitable approach in controlling and managing quality in web based application.

Course Content:

Unit-1: Web-Based Systems**,** Web Applications, WebApps—A Philosophical View; Web Engineering: What Is Web Engineering?, The Components of Web Engineering, Web Engineering Best Practices; Communication: The Communication Activity, Formulation, Elicitation, Identifying WebApp Increments, Negotiation; Planning: Understanding Scope, Refining Framework Activities, Building a WebE Team,

Unit-2: The Modeling Activity: Modeling as a Concept, The Models We Create, Modeling Frameworks, Modeling Languages, Existing Modeling Approaches; Analysis Modeling for WebApps: Understanding Analysis in the Context of WebE, Analysis Modeling for WebApps, Understanding the Users.

Unit-3: Construction and Deployment: Construction and Deployment within the WebE Process, Construction Principles and Concepts, Deployment, Construction and the Use of Components, Component-Level Design Guidelines, Component Design Steps; Testing WebApps: Testing Concepts, The Testing Process—An Overview, Content Testing, User Interface Testing, Usability Testing, Compatibility Testing, Component-Level Testing, Navigation Testing, Configuration Testing, Security and Performance Testing.

Unit-4: The ISO9000 series of quality management standards: The purpose of standards, the ISO9000 series: a generic quality management standard, ISO9000-3: notes for guidance on the application of ISO9001 in software development, the impact of ISO9000 and TickIT. Models and standards for process improvement: The Capability Maturity Model, individual levels of CMM, the role of the CMM, SPICE modeling.

Unit-5: Tools for Quality Improvement: basic quality control tools, check sheet, cause and effect diagram, pareto diagram, histogram, scatter plot, run chart, control chart, orthogonal defect classification.

. **Evaluation Scheme:**

Evaluations	Marks	Remarks
T-1	15 Marks (1-Hours)	1st - 4th Week
T-2	25 Marks (1:30 Hours)	5th - 10th Week
T-3	35 Marks (2-Hours)	11th - 16th Week
Assignments	10 Marks	
Tutorials / Subject Seminar	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Teaching Methodology:

The course will be delivered through lecture oriented towards understanding and designing of web pages using the web tools. It will impart strong foundation of Web Application Terminologies, Internet Tools, E – Commerce and other web services.

Text Books

1. Web Engineering: A Practitioner's Approach/ Roger Pressman, David Lowe.
2. Web Technologies: Achyut Godbole, Atul Kahate, McGraw-Hill Publications (Third Edition).
3. Beginning Android Application Development, Wei-Meng Lee, Wrox.

Reference Books

1. Internet & World Wide Web How to Program / Deitel, H.M.
2. Web Design with HTML/Flash/Java Script and E-Commerce Bible / Crowder, David
3. Database Driven Web Sites / Feiler, Jesse
4. Web design: the complete reference / Powell Thomas A
5. Internet 101: a beginner's guide to the Internet and the WorldWideWeb/Lehnert Wendy G
6. E-Commerce: Fundamentals and Applications / Chan, Henry
7. E-commerce: strategy, technology & applications / Whiteley, David
8. E-Commerce Logistics & Fulfillment: delivering the goods / Bayles, Deborah L.

Web References

1. www.w3schools.com
2. http://www.techtutorials.info/ecommerce.html

Journals

1. ACM Transactions on the Web (TWEB).
2. ACM Transactions on the Information Systems (TOIC).
3. ACM Transactions on Graphics (TOG).
4. ACM Transactions on Internet Technology (TOIT).

Title of Course: Full Stack Development **Course Code: CS311**
L-T-P scheme: 3-0-0 **Credit: 3**

Prerequisite: Students must have already registered for the course, "***Introduction to Computers and Programming***" and "**Object Oriented Programming**".

Objective:

1. To learn and be able to implement the front-end and back-end web-technologies.
2. To develop the abilities to call oneself full-stack web developer.

Learning Outcomes:

At the end of the course, students will:

1. Get **familiar** with process of full stack Web Development.
2. Have a good grounding of Web Application Terminologies, Internet tools and languages like HTML5 and CSS, and identify the typical use cases where to **apply** these tools.
   - **Analyze** a problem and possess demonstrative skills in using and applying JavaScript to provide solutions.
   - **Design and code** the business requirements to come up with a technical solution using different web-based technologies including front-end and back-end frameworks; databases like MySQL and MongoDB.
   - Work as a team on a project.

CO	PO [As per NBA]
CO1	
CO2	PO1
CO3	PO2
CO4	PO3
CO5	PO5

Course Content:

Part-1: Fundamentals of Web Development

Unit-1 Creating first web-application, hosting a web application, creating websites, authoring tools, domain names. architectures.

Part-2: Front End Tools & Technologies

Unit-2 Markup and Styling: HTML, Cascading Style Sheets, using Bootstrap.

Unit-3 JavaScript Fundamentals: Language Features, JSON, Ajax, jQuery, Popular Frameworks like React, Angular JS.

Part-3: Back End Tools & Technologies

Unit-4 Web Programming through Node.js and/or Java. Node.js Modules, NPM, Events, Upload File, Email, Get/Post methods, Java Servlets vs. JSP, Request vs. Response objects, other Java objects and features.

Unit-5 Databases and Web Storage: Designing and creating databases, database connection through back-end programming languages, Web storage to store sessions, cookies, and cached data in the browser.

Part-4: Miscellaneous

Unit-6 HTTP & REST, RESTful API, Chrome DevTools, SSL Certificates, Web Application Architecture, MVC, Platforms as a service, Heroku and AWS, Web Security.

Unit-7 Git, Common git commands, Data Structures & Algorithms, Understanding hash tables, trees, graphs, Big-O analysis, object vs an array, pros and cons of in-memory vs disk storage, difference between queues and stacks.

Teaching Methodology:

This course is introduced to help students transition from a simple developer to a full stack developer. Starting from frontend development, the student will slowly progress to become to other aspects of development including backend, database, version control and other essential technologies that are helpful for a developer. The entire course is broken down into four separate parts: Fundamentals of Web Development, Front End tools & Technologies, Back End Tools & Technologies, and Project Development. Each section includes multiple technologies to help a student gain more experience as a developer. This theory course is well complemented by a lab course under the name Full Stack Development Lab in the same semester that helps a student learn and discuss the technical details of the underlying technologies.

Evaluation Scheme:

Exams		**Marks**	**Coverage**
T-1		15 Marks	**Based on Units: 1-2**
T-2		25 Marks	**Based on Units: 1-4**
T-3		35 Marks	**Based on Units:1-6**
Teacher's Assessment	Assignment	10 Marks	**25 Marks**
	Tutorial	05 Marks	
	Quiz	05 Marks	
	Attendance	05 Marks	
Total		**100 Marks**	

Learning Resources:

Tutorials and lecture slides on Full Stack Development (will be added from time to time): Digital copy will be available on the JUET server/ Google Classroom.

Text Book

1. Web Technologies: Achyut Godbole, Atul Kahate, McGraw-Hill Education (Third Edition).
2. Web Engineering: A Practitioner's Approach by Roger Pressman and David Lowe, McGraw-Hill, 2009.
3. HTML and CSS: Comprehensive 7th edition, by by Denise M. Woods and William J. Dorin. Publisher: Cengage Learning; (2012) ISBN-10:1133526144

4. Internet & World Wide Web How to Program, 5/e Paul J. Deitel, Harvey M. Deitel, Abbey Deitel, Pearson Education 2012.

Reference Books

1. Internet & World Wide Web How to Program / Deitel, H.M.
2. Web Design with HTML/Flash/Java Script and E-Commerce Bible / Crowder, David
3. Database Driven Web Sites / Feiler, Jesse
4. Web design: the complete reference / Powell Thomas A
5. Internet 101: a beginner's guide to the Internet and the WorldWideWeb/Lehnert Wendy G
6. E-Commerce: Fundamentals and Applications / Chan, Henry
7. E-commerce: strategy, technology & applications / Whiteley, David
8. E-Commerce Logistics & Fulfillment: delivering the goods / Bayles, Deborah L.

Web References:

1. www.w3schools.com
2. http://www.techtutorials.info/ecommerce.html

Journals:

1. ACM Transactions on the Web (TWEB).
2. ACM Transactions on the Information Systems (TOIC).
3. ACM Transactions on Graphics (TOG).
4. ACM Transactions on Internet Technology (TOIT).

Course Name: Mobile Application Development **Course Code:** CS312
L-T-P scheme: 3-0-0 **Credits:** 3

Prerequisite:
Students should have knowledge of object-oriented programming and basic web development. Familiarity with Java or Kotlin is recommended.

Objective:

This course is aimed:

- To introduce students to the principles and practices of mobile app development.
- To provide hands-on experience with mobile development platforms such as Android.
- To equip students with the ability to design, implement, test, and publish mobile apps.
- To expose students to real-world constraints like battery, performance, security, and user interface design on mobile devices.

Learning Outcomes:

Course Outcome	Description
CO1	Understand the architecture and components of mobile platforms, especially Android.
CO2	Design user interfaces using XML and create responsive layouts for various screen sizes.
CO3	Develop interactive applications using activities, intents, fragments, and services.
CO4	Handle data using SQLite, Room, SharedPreferences, and connect to RESTful web services.
CO5	Implement mobile-specific features such as sensors, location services, camera, and notifications.
CO6	Apply best practices for testing, debugging, deploying, and securing mobile applications.

Course Contents:

Unit 1: Introduction to Mobile Development
Overview of mobile platforms, Android architecture and components. Activity lifecycle. Development tools: Android Studio, AVD, SDK Manager. Manifest file and application structure.

Unit 2: User Interface Design and Navigation
View hierarchy and layout design using XML. Widgets, styles, themes, event handling. Navigation components, toolbar, menus, recycler view. Supporting multiple screens and orientations.

Unit 3: Application Components
Activities, fragments, intents, services, broadcast receivers. Inter-component communication. Lifecycle methods and callbacks. Background processing with AsyncTask, Handler, and WorkManager.

Unit 4: Data Management and Networking
SQLite database and Room persistence library. SharedPreferences and file I/O. JSON parsing, REST

API integration using Retrofit/Volley. Consuming remote data and managing background network calls.

Unit 5: Device Features and Advanced Topics
Using sensors, camera, audio, and video APIs. Location and mapping with Google Maps API. Notifications and background services. App permissions and manifest declarations. Security and battery considerations.

Unit 6: App Testing, Debugging, and Deployment
Debugging tools in Android Studio. Unit testing and UI testing. Preparing apps for release. APK generation and publishing on Google Play. Best practices in mobile app performance and security.

Methodology:

The course includes theory lectures, tool demonstrations, and guided mini-projects. Assignments will focus on app design and implementation in Android Studio. Emphasis is placed on UI/UX, data handling, and integration of mobile features. Students will develop and present a fully working mobile application by the end of the course.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered up to Test-1
Test-2	25 Marks	Syllabus covered up to Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Books:

1. Dawn Griffiths and David Griffiths, *Head First Android Development*, O'Reilly.
2. Joseph Annuzzi, Lauren Darcey, and Shane Conder, *Android Programming: The Big Nerd Ranch Guide*, Pearson.
3. Neil Smyth, *Android Studio Development Essentials*, Techotopia.
4. Jeff Friesen, *Learn Java for Android Development*, Apress.
5. Android Developer Documentation – https://developer.android.com/

Course Name: Natural Language Processing **Course Code:** CS313
L-T-P scheme: 3-0-0 **Credits:** 3

Prerequisite:
Students should be familiar with probability, linear algebra, algorithms, and basic programming (preferably in Python).

Objective:

This course is aimed:

- To introduce the foundational concepts and algorithms used in Natural Language Processing (NLP).
- To explore computational techniques for lexical, syntactic, semantic, and pragmatic analysis of natural language.
- To enable students to implement and evaluate models using real-world text data for practical applications such as classification, sentiment analysis, and translation.
- To expose students to modern advances in deep learning-based NLP such as transformers and contextual embeddings.

Learning Outcomes:

Course Outcome	Description
CO1	Understand the core NLP pipeline and preprocessing techniques for text normalization and structure.
CO2	Apply statistical models and language models to tasks like part-of-speech tagging and sentence probability estimation.
CO3	Analyze syntactic structures of sentences using parsing algorithms and grammars.
CO4	Develop and use semantic representations of words and sentences for similarity and inference tasks.
CO5	Build and evaluate classification models for text-based applications such as sentiment analysis and spam detection.
CO6	Utilize neural networks and transformer architectures for advanced NLP tasks such as language modeling and sequence labeling.

Course Contents:

Unit 1: Introduction to NLP and Text Processing
This unit introduces the field of Natural Language Processing and its applications in areas such as machine translation, sentiment analysis, and information retrieval. It covers the structure of language (morphology, syntax, semantics), the components of the NLP pipeline, and common text preprocessing techniques such as tokenization, lowercasing, stopword removal, stemming, and lemmatization. Students will also learn to use corpora and implement basic cleaning and transformation steps using libraries like NLTK and spaCy.

Unit 2: Statistical Language Models and Part-of-Speech Tagging
Students will study how statistical models can be used to represent and predict natural language. The

unit covers n-gram models and their probability estimation, smoothing techniques (Laplace, Good-Turing), and evaluation with perplexity. It introduces POS tagging as a sequence labeling problem and covers methods such as rule-based tagging, statistical tagging using Hidden Markov Models (HMMs), and the Viterbi decoding algorithm.

Unit 3: Syntax and Parsing Techniques
This unit covers syntactic analysis of sentences using grammar-based methods. It includes context-free grammars (CFGs) for representing language structure and parsing algorithms such as top-down and bottom-up approaches, including the CYK algorithm. Students will also explore dependency parsing to analyze grammatical relations between words and learn to use syntactic parsing tools available in libraries like spaCy and Stanford NLP.

Unit 4: Semantic Representations and Word Embeddings
In this unit, students will learn how to capture meaning through vector-based models. It covers the Bag-of-Words model, TF-IDF, and distributional semantics. Neural embedding techniques such as Word2Vec (CBOW and Skip-Gram) and GloVe will be introduced. The unit will also discuss sentence representations, semantic similarity, and the fundamentals of contextual word embeddings, including an overview of BERT.

Unit 5: Text Classification and NLP Applications
This unit focuses on text classification tasks and real-world NLP use cases. Students will explore supervised learning models such as Naive Bayes, Logistic Regression, and Support Vector Machines applied to tasks like spam detection and sentiment analysis. The unit includes feature engineering with n-grams and TF-IDF, and performance evaluation using accuracy, precision, recall, and F1-score. Students will also be introduced to unsupervised techniques like Latent Dirichlet Allocation (LDA) for topic modeling, and basic Named Entity Recognition (NER).

Unit 6: Deep Learning and Transformers in NLP
The final unit introduces modern neural network-based approaches for sequence modeling and language understanding. Students will study Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and attention mechanisms. The unit progresses into the Transformer architecture, explaining components like self-attention and positional encoding. It concludes with the use of pretrained models such as BERT and GPT for downstream NLP tasks.

Methodology:

The course is delivered through a combination of lectures and practical sessions. Each concept is accompanied by demonstrations using Python and standard NLP libraries. Weekly assignments and mini-projects will help students gain hands-on experience in building and evaluating NLP systems. Interactive quizzes and in-class activities will reinforce theoretical learning.

Evaluation Scheme:

Component	Marks	Coverage
Test-1	15 Marks	Topics covered up to Test-1
Test-2	25 Marks	Topics covered up to Test-2
Test-3	35 Marks	Full syllabus
Assignment	10 Marks	Weekly coding and theory tasks

Tutorials	5 Marks	In-class practice sessions
Quiz	5 Marks	Objective evaluations
Attendance	5 Marks	Based on student presence
Total	**100 Marks**	

Learning Resources:

1. **Daniel Jurafsky and James H. Martin**, *Speech and Language Processing*, Pearson – A comprehensive textbook covering both traditional and modern NLP techniques.
2. **Steven Bird, Ewan Klein, and Edward Loper**, *Natural Language Processing with Python*, O'Reilly – Practical guide using Python and NLTK for hands-on NLP tasks.
3. **Yoav Goldberg**, *Neural Network Methods for Natural Language Processing*, Morgan & Claypool – A focused text on deep learning approaches in NLP.
4. **Delip Rao and Brian McMahan**, *Natural Language Processing in Action*, Manning – Application-oriented book using real-world examples.
5. **Hugging Face Transformers Library** – Official documentation and tutorials for using pretrained transformer models: https://huggingface.co/docs
6. **NLTK and spaCy Documentation** – References for popular Python NLP libraries:
    - NLTK: https://www.nltk.org
    - spaCy: https://spacy.io

Title: Soft Computing **Course Code: CS314**

L-T-P Scheme: 3-0-0 **Credit: 3**

Pre-requisite: Artificial Intelligence & Application

Course Objectives:

This course aims to develop students' abilities in using some contemporary approaches in solving problems in automation.

It will enable students to:
(a) Appreciate the advantages and limitations of fuzzy systems and their potential impacts and applications in intelligent control and automation;
(b) Appreciate the advantages and limitations of neural networks and their potential impacts and applications in intelligent automation; and
(c) Develop an understanding of generic algorithms and their potential applications.

Learning Outcomes:

After completing this course, you will be able to learn:

- Fuzzy logic and its applications.
- Artificial neural networks and its applications.
- Solving single-objective optimization problems using GAs.
- Solving multi-objective optimization problems using Evolutionary algorithms (MOEAs).
- Applications of Soft computing to solve problems in varieties of application domains.

Course Outcome	Description
CO1	Demonstrate knowledge of the building blocks of Soft Computing as presented in terms of intelligent agents.
CO2	Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game based techniques to solve them.
CO3	Develop algorithms for real life problems problems and also design intelligent systems.
CO4	Attain the capability to represent various real life problem domains using fuzzy logic, Artificial Neural Network and Genetic Algorithms based techniques.
CO5	Formulate and solve problems with uncertain information using Soft Computing approaches.
CO6	Apply concept of Soft Computing for processing to problems leading to understanding of cognitive computing.

Course Contents:
Basics of Soft Computing
Fundamental of Neural Networks
Back-propagation Networks

Associative Memory
Adaptive Resonance Theory
Fuzzy Set Theory
Fuzzy Systems
Fundamentals of Genetic Algorithms
Genetic Modeling
Integration of Neural Networks, Fuzzy Logic, and Genetic Algorithms

Learning Resources:
Lecture presentations, assignments will be posted on the student resource from time to time. In addition following additional online/downloadable resources will be useful.

Text Book:
1. "Neural Networks, Fuzzy Logic, and Genetic Algorithms" by **S. Rajasekaran, G.A. Vijayalakshmi Pai,** (Prentice-Hall of India Private Ltd.),

Other References:

1. "Neuro-Fuzzy And Soft Computing" by J. S. R. Jang, C. T. Sun, E. Mizutani (Pearson Education)
2. "Soft Computing in Human-Related Science" by Horia-Nicolai Teodorescu, Abraham Kandel, Lakhmi C. Jain (CRC Press)
3. "Genetic Algorithms" by David E. Goldberg (Pearson Education)
4. "Soft Computing and Intelligent Systems: theory and Application" by Sinha, Naresh K.
5. "Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence" by Kosko, Bart

Course Title: Data Science **Course Code: CS315**
L-T-P Scheme: 3-0-0 **Credit: 3**

Prerequisite: Students must have already studied the course "Business Analysis Techniques"

Course Objectives:
This course will introduce students to this rapidly growing field and equip them with some of its basic principles and tools as well as its general mindset. Students will learn concepts, techniques and tools they need to deal with various facets of data Science practice, including data collection and integration, exploratory data analysis, predictive modeling, descriptive modeling, data product creation, evaluation, and effective communication.

Learning Outcomes:
At the end of the course students should be able to:

Course Outcome	Description
CO1	Develop relevant programming abilities.
CO2	Demonstrate proficiency with statistical analysis of data.
CO3	Develop the ability to build and assess data-based models.
CO4	Execute statistical analyses with professional statistical software.
CO5	Demonstrate skill in data management.
CO6	apply data science concepts and methods to solve problems in real-world contexts

Course Content:

Unit I: Introductionand Data Pre-processing
Data Science Introduction, Big Data and Data Science, Current landscape of perspectives

Unit II: Data Analysis and Correlations: Basic Concepts and Methods
Populations and samples, Statistical modelling, probability distributions, Regression, fitting a model Dimensionality Reduction: PCA & DWT, Correlation and regression analysis. Chisquare t and F distributions (definitions only) Confidence interval Single mean and difference known and unknown variances.

Unit III: Introduction to machine learning and Cluster Analysis: Basic Concept and Methods
Supervised and unsupervised learning, Training and testing data, over fitting and under fitting. Distance measures :- Manhattan, Chebbychev, Mahalanobis Distance, Cluster Analysis, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid-Based Methods, Evaluation of Clustering, Clustering High-Dimensional Data, Clustering Graph and Network Data

Unit IV: Classification Algorithms
Basic Concepts, Decision Tree Induction, Bayes Classification Methods, Rule-Based Classification, Model Evaluation and Selection, Techniques to Improve Classification Accuracy, Support Vector Machines, Lazy Learners (or Learning from Your Neighbors)

Unit V: Introduction to Web Search and Social Media Analytics

Data Wrangling: APIs and other tools for scrapping the Web Mining Complex Data Types, Other Methodologies of Data, Mining, Data Mining Applications, Data Mining and Society, Data Mining

Trends Social Media Analytics is the science of analyzing data to convert information to useful knowledge. This knowledge could help us understand our world better and, in many contexts, enable us to make better decisions.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Based on Unit-1, Unit-2
Test-2	25 Marks	Based on Unit-3 & Unit-4 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-5 around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Text Books:

1. Cathy O'Neil and Rachel Schutt. Doing Data Science, Straight Talk fromthe Frontline. O'Reilly. 2014.
2. Avrim Blum, John Hopcroft and Ravindran Kannan. Foundations of Data Science.

Refernce Books:

1. Jure Leskovek, Anand Rajaraman and Je_rey Ullman. Mining of Massive Datasets. v2.1,
2. Cambridge University Press. 2014.
3. Kevin P. Murphy. Machine Learning: A Probabilistic Perspective. ISBN 0262018020. 2013.
4. Foster Provost and Tom Fawcett. Data Science for Business: What You Need to Know aboutData Mining and Data-analytic Thinking. ISBN 1449361323. 2013.
5. Trevor Hastie, Robert Tibshirani and Jerome Friedman. Elements of Statistical Learning, Second Edition. ISBN 0387952845. 2009.

Course Name: Digital Signal Processing **Course Code:** CS316
L-T-P scheme: 3-0-0 **Credits:** 3

Prerequisite:
Basic knowledge of Signals and Systems, discrete-time signals, Fourier analysis, and calculus.

Objective:

This course aims to introduce students to the theoretical foundations and practical aspects of digital signal processing. It focuses on the analysis and manipulation of discrete-time signals and systems using techniques such as z-transforms, Fourier transforms, and digital filter design. Students will gain exposure to DSP algorithms and their implementation in real-time applications.

Learning Outcomes:

Course Outcome	Description
CO1	Understand discrete-time signals, systems, and their classification.
CO2	Analyze discrete signals using z-transforms and discrete-time Fourier techniques.
CO3	Apply convolution and correlation for system analysis and filtering.
CO4	Design FIR and IIR digital filters using classical and modern techniques.
CO5	Implement and evaluate DSP algorithms for real-time applications using software tools.
CO6	Understand applications of DSP in areas such as audio processing, image processing, and communications.

Course Contents:

Unit 1: Discrete-Time Signals and Systems
This unit introduces various types of discrete-time signals such as unit step, unit impulse, exponential, and sinusoidal sequences. Classification of systems (linear, time-invariant, causal, stable) is discussed along with system properties. Linear convolution and the response of LTI systems to arbitrary inputs are analyzed.

Unit 2: Z-Transform and Its Applications
Z-transform and its properties are covered, including ROC (region of convergence), inverse z-transform techniques, and system function representation. Analysis of linear time-invariant systems in the z-domain and system stability using pole-zero plots are discussed.

Unit 3: Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT)
The unit covers Discrete Fourier Transform (DFT), its properties, and computation. Efficient computation of DFT using FFT algorithms such as Radix-2 Decimation-in-Time (DIT) and Decimation-in-Frequency (DIF) are explained. Applications in spectrum analysis are discussed.

Unit 4: FIR and IIR Digital Filter Design
This unit introduces the design and analysis of digital filters. FIR filter design using windowing and frequency sampling techniques is discussed. IIR filter design from analog prototypes using methods such as bilinear transformation and impulse invariance is covered. Filter specifications, frequency response, and phase considerations are included.

Unit 5: Implementation of DSP Systems
The final unit focuses on practical implementation aspects of DSP. Structures for FIR and IIR filter realization such as direct form, cascade, and parallel forms are discussed. Fixed-point and floating-point implementation issues, DSP processor features, and application areas such as audio and biomedical signal processing are introduced.

Methodology:
The course will be delivered through interactive lectures and problem-solving sessions. MATLAB and Python will be used for algorithm implementation and simulation of DSP systems. Assignments and mini-projects will provide hands-on experience. Emphasis will be given to visualization of signals and filters using computational tools.

Evaluation Scheme:

Component	Marks	Coverage
Test-1	15 Marks	Topics covered up to Test-1
Test-2	25 Marks	Topics covered up to Test-2
Test-3	35 Marks	Full syllabus
Assignment	10 Marks	Weekly coding and theory tasks
Tutorials	5 Marks	In-class practice sessions
Quiz	5 Marks	Objective evaluations
Attendance	5 Marks	Based on student presence
Total	**100 Marks**	

Learning Resources:

1. **Alan V. Oppenheim and Ronald W. Schafer**, *Discrete-Time Signal Processing*, Pearson – A foundational text covering theory and application of DSP.
2. **John G. Proakis and Dimitris G. Manolakis**, *Digital Signal Processing: Principles, Algorithms and Applications*, Pearson – Widely used for filter design and DSP algorithms.
3. **Sanjit K. Mitra**, *Digital Signal Processing: A Computer-Based Approach*, McGraw-Hill – Practical approach with MATLAB-based examples.
4. **Richard G. Lyons**, *Understanding Digital Signal Processing*, Pearson – Conceptual and intuitive coverage with hands-on examples.
5. **MATLAB Documentation** – https://www.mathworks.com/help
6. **Python DSP Libraries** – SciPy and NumPy documentation: https://docs.scipy.org/doc

Course Name: Statistical Inference **Course Code:** CS317
L-T-P scheme: 3-0-0 **Credits:** 3

Prerequisite:

Students should have completed a basic course in probability theory and mathematical statistics and be comfortable with calculus and basic linear algebra.

Objective:

- To develop a solid foundation in statistical inference, both theoretical and applied.
- To introduce methods for estimating parameters, testing hypotheses, and constructing confidence intervals.
- To prepare students for research or professional work involving statistical modeling and data analysis.

Learning Outcomes:

Course Outcome	Description
CO1	Understand the theoretical underpinnings of estimation, including properties like unbiasedness, consistency, and efficiency.
CO2	Apply point and interval estimation methods including MLE, method of moments, and Bayesian estimators.
CO3	Analyze and design hypothesis tests using classical approaches like Neyman–Pearson and likelihood ratio tests.
CO4	Use sampling distributions to draw conclusions about population parameters.
CO5	Apply concepts of sufficiency, completeness, and exponential families in the context of estimation and testing.
CO6	Utilize nonparametric tests and assess their suitability for different types of data.

Course Contents:

Unit 1: Probability and Sampling Distributions

Review of probability theory, convergence concepts, law of large numbers, central limit theorem, common sampling distributions (Normal, Chi-square, t, F).

Unit 2: Point Estimation

Estimators and their properties (bias, consistency, efficiency), sufficiency and completeness, MVUE, Rao-Blackwell and Lehmann–Scheffé theorems, Cramér–Rao lower bound, method of moments, maximum likelihood estimation.

Unit 3: Interval Estimation

Confidence intervals for population parameters, pivotal quantities, intervals based on MLE and asymptotic results.

Unit 4: Hypothesis Testing

Basic concepts, Type I and II errors, power, critical regions, Neyman–Pearson lemma, likelihood ratio tests, UMP and UMPU tests, p-values.

Unit 5: Nonparametric Tests

Introduction to nonparametric methods including sign test, Wilcoxon signed-rank test, rank-sum test, and applications.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Syllabus covered up to Test-1
Test-2	25 Marks	Syllabus covered up to Test-2
Test-3	35 Marks	Full Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Books:

1. George Casella and Roger L. Berger, *Statistical Inference* (2nd Edition), Duxbury, 2001.
2. Jun Shao, *Mathematical Statistics* (2nd Edition), Springer, 2007.
3. V. K. Rohatgi and A. K. Md. Ehsanes Saleh, *An Introduction to Probability and Statistics* (2nd Edition), Wiley, 2000.
4. B. L. S. Prakasa Rao, *A First Course in Probability and Statistics*, World Scientific, 2009.
5. George Roussas, *An Introduction to Probability and Statistical Inference*, Academic Press, 2002.
6. Kai Lai Chung, *A Course in Probability Theory* (3rd Edition), Academic Press, 2001.

Title of Course: Information Retrieval Systems **Course Code:** CS318
L-T Scheme: 3-0-0 **Course Credits:** 3

Objectives:
This course focuses on the design and implementation of information retrieval systems for unstructured data, including techniques for indexing, ranking, web crawling, and semantic web technologies such as ontologies and RDF. Students will learn to analyze models and metrics used in evaluating IR systems, and understand how to apply these technologies in web-based environments.

Learning Outcomes:
Students will gain the ability to develop and evaluate IR systems, understand the mechanics of search engines, web crawling, and the semantic structure of the web. They'll also be able to apply techniques such as query processing, term weighting, and ontology creation to real-world datasets.

Course Outcome

Course Outcome	Description
CO1	Design and implement information retrieval systems for unstructured data.
CO2	Apply query processing techniques for tolerant retrieval.
CO3	Analyze information retrieval models and their metrics.
CO4	Analyze the searching algorithms for Information Retrieval.
CO5	Demonstrate the web crawling, taxonomy and ontology of web applications.

Course Contents:

Unit 1: Introduction to Information Retrieval
Theory of information retrieval, information retrieval on data and information retrieval on the web. Information retrieval tools and their architecture.

Unit 2: Boolean Retrieval & Index Construction
An example information retrieval problem, processing Boolean queries, the extended Boolean model versus ranked retrieval, blocked sort-based indexing, single pass in-memory indexing, distributed and dynamic indexing.

Unit 3: Dictionary and Tolerant Retrieval
Wildcard queries, spelling correction, phonetic correction.

Unit 4: Scoring, Term Weighting and the Vector Space Model
Term frequency and weighting, vector space model, variant TF-IDF scoring, probabilistic model, evaluation of IR system.

Unit 6: Information Retrieval Tools
Web directory, search engine, meta-search engines, web searching and search engine architecture, searching algorithms (Fish, Shark, etc.), and PageRank algorithms.

Project-Based Learning:
Each student, in a group of 3–4, will choose an issue related to an information retrieval system. Each group will identify recent research related to the problem area. The group will analyze the solution proposed in the articles and implement it on a real dataset.

Text Books:

1. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *An Introduction to Information Retrieval*, 2013, Cambridge University Press.
2. C. J. van Rijsbergen, *Information Retrieval*, 2nd Edition, 2012.

Reference Books:

1. Salton, G. and McGill, M. J., *Introduction to Modern Information Retrieval*, Computer Series, McGraw-Hill, New York, NY.
2. *ACM Transactions on Internet Technology*

Course Name: Augmented and Virtual Reality
Course Code: CS320
L-T-P Scheme: 3-0-0
Credits: 3

Prerequisite:
Basic knowledge of computer graphics, programming, and human-computer interaction.

Objective:

This course aims:

- To introduce the fundamental concepts of Augmented Reality (AR), Virtual Reality (VR), and Mixed Reality (MR).
- To explore hardware, software, and interaction techniques used in immersive environments.
- To provide practical exposure to VR/AR development tools and technologies.
- To demonstrate the applications of XR (Extended Reality) in entertainment, simulation, and other real-world scenarios.

Learning Outcomes:

Course Outcome	Description
CO1	Understand the core principles and differences among AR, VR, and MR technologies.
CO2	Analyze the hardware architecture and components of immersive systems.
CO3	Explain stereoscopic vision, haptic feedback, and their integration in XR systems.
CO4	Design and develop VR/AR applications using modern SDKs and development platforms.
CO5	Apply 3D interaction techniques in immersive environments.
CO6	Demonstrate real-world use cases of AR/VR in digital entertainment and simulation.

Course Contents:

Unit 1: Introduction to Extended Reality (XR), Introduction to AR, VR, and MR**,** Taxonomy and key features of AR systems**,** Differences between AR, VR, and MR**,** Challenges in AR technology**,** AR system functionality and visualization techniques.

Unit 2: VR Systems and Hardware, VR as a discipline and core features, Architecture of VR systems, VR input hardware: tracking systems, motion capture, data gloves, VR output hardware: visual displays and rendering.

Unit 3: Human Perception, Stereoscopic Vision & Haptics, Human visual system fundamentals and depth cues**,** Stereopsis and retinal disparity**,** Haptic perception and devices**,** Algorithms for haptic rendering and stereo image synthesis.

Unit 4: XR Software Development, Challenges in VR software development, System architectures: Master/Slave, Client/Server, Cluster rendering in VR, Game engines and SDKs: Unity, Unreal, ARToolkit, Developing for HTC Vive, Oculus, Google VR, AR concepts: camera calibration, marker-based AR

Unit 5: 3D Interaction & XR Applications, 3D manipulation tasks and input devices, Interaction techniques for immersive systems, Applications in digital entertainment: film, TV, fitness, gaming, Demonstration of VR-based digital content

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15	Syllabus covered up to Test-1
Test-2	25	Syllabus covered up to Test-2
Test-3	35	Full syllabus
Assignment	10	
Tutorials	5	
Quiz	5	
Attendance	5	
Total	**100**	

Learning Resources:

Books:

1. Alan B. Craig, *Understanding Augmented Reality*, Morgan Kaufmann.
2. Grigore C. Burdea, Philippe Coiffet, *Virtual Reality Technology*, Wiley.
3. Dieter Schmalstieg, Tobias Hollerer, *Augmented Reality: Principles and Practice*, Addison-Wesley.
4. Jason Jerald, *The VR Book: Human-Centered Design for Virtual Reality*, ACM Books.
5. Joseph L. Gabbard, *A Survey of User-Centered Evaluations in AR Systems*, IEEE Transactions on AR.

Title of Course: Data Mining **Course Code: CS321**
L-T Scheme: 3-0-0 **Course Credits: 3**

Objectives: To study advanced aspects of data warehousing and data mining, encompassing the principles, research results and commercial application of the technologies.

Learning Outcomes:
At the end of the course the students will have knowledge of:

Course Outcome	Description
CO1	Data analysis methods, covering traditional methods but with greater emphasis on modern methods that locate and address common data foibles
CO2	Survey design & data collection issues
CO3	Multivariate methods: supervised/unsupervised classification, data reduction
CO4	Univariate methods: both basic (e.g. t-tests, ANOVA, linear models) and advanced (e.g. Generalized Linear Models, Generalized Additive Models).
CO5	Data mining methods: tree methods with boosting and bagging; Multivariate Adaptive Regression Splines; Random Forests; Neural Nets; model diagnostics
CO6	Tools for difficult data: ridge regression; basic data imputation

Course Contents:

Unit-I: Data Mining and Knowledge Discovery, The KDD process and methodology, Data preparation for knowledge discovery, Overview of data mining and Machine Learning techniques, Review of Python and overview of Python tools for Data Analysis.

Unit-II: Supervised Techniques, Classification and Prediction using K-Nearest-Neighbor, Classifying with Probability Theory; Naïve Bayes, Building Decision Trees, Forecasting and Regression models, Evaluating predictive models.

Unit-III: Unsupervised Learning, Clustering using K-Means, Association Rule discovery, Sequential Pattern Analysis, Principal Component Analysis and Dimensionality, Reduction.

Unit-IV: Possible Applications (covered throughout the course), Collaborative Recommender Systems, Content Based personalization, Predictive User Modeling, Concept Discovery from Documents, Blogs, Social Annotations, Finding groups using social or behavioral data, Building predictive models for target marketing, Customer or user segmentation.

Unit-V: Advance Topics (if time permits), SVD and Matrix Factorization, Search and Optimization Techniques, Markov Models, Dealing with Big Data and Map Reduce.

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Syllabus covered upto T-1
Test-2	25 Marks	Syllabus covered upto T-2
Test-3	35 Marks	Entire Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Text Books

1. "Building the Data Warehouse", W. H. Inman, 3rd edition, John Wiley & Sons.
2. "Data Mining Techniques", Arun K. Pujari, University Press.

References

1. W.H.Inmon, C.L.Gassey, "Managing the Data Warehouse", John Wiley & Sons.
2. Fayyad, Usama M. et. al., "Advances in knowledge discovery & Data-Mining", MIT Press.
3. Dunham, Margaret H.,"Data Mining –Introductory and Advanced Topics.

Title of Course: Agile and Secure Software Engineering **Course Code: CS322**
L-T Scheme: 3-0-0 **Course Credits: 3**

Prerequisite: Students must have already registered for the course, "Software Engineering / Software Development Fundamentals.

Objectives:

1. Understand the principles of agile software development and its methodologies.
2. Learn how to integrate security throughout the agile software development lifecycle (SDLC).
3. Apply secure coding practices and threat modeling within agile teams.
4. Explore automation and tooling (e.g., CI/CD, static/dynamic analysis) to ensure continuous security.
5. Evaluate agile methods for effectiveness in delivering secure, high-quality software.

Learning Outcomes:
At the end of the course the students will have knowledge of:

Course Outcome	Description
CO1	Interpret the trade-offs between traditional software development methods and agile software development methods for a software project effectively
CO2	Identify and make use of an appropriate agile software engineering approach viz. extreme programming, Scrum, Crystal techniques as a part of software development.
CO3	Apply Refactoring techniques on source code for improved design
CO4	Choose tools and construct the methods for testing Agile projects using various testing strategies
CO5	List the Planning, tracking, estimation and monitoring of agile projects with techniques like burn down charts, velocity calculation and task boards etc.

Course Contents:

Unit-I: Introduction
Traditional software development methods, Agile software development methods and lean software development methods

Unit-II: Agile Fundamentals, Requirements and Planning
Agile manifesto, Agile principles, Characteristics of Agile processes, an iterative development process, Pros and cons of incremental development and software prototyping. User stories, agile estimation, planning techniquesPrioritizing Themes, Financial prioritization, prioritizing desirability

Unit-III: Agile Models
Introduction, Scrum - Prioritizing, Estimating, and Planning, The Scrum Experience (hands-on exercise), Extreme Programming Values, Principles and Practices, Pair programming, Embracing change, incremental change, Crystal methodologies: project categories, complexity, family members, Crystal's seven properties, Crystal clear development process cycle, Crystal yellow, crystal orange and crystal orange web, The principles of kanban, Improving process with kanban, Measure and manage flow, Emergent behavior, Processes of feature driven development, practices and progress in FDD.

Unit-IV: Testing and Refactoring

Agile testing strategy, automated unit test, test plan, test driven development, alpha, beta and acceptance testing, Bad smells in code, properties of refactoring, refactoring examples, benefits, cost and risk of refactoring

Unit-V: Secure Software Engineering Concepts

Principles of Secure Design, OWASP Top 10 and Common Vulnerabilities, Secure Coding Practices, Security Testing Techniques

Evaluation Scheme:

Exams	Marks	Coverage
Test-1	15 Marks	Syllabus covered upto T-1
Test-2	25 Marks	Syllabus covered upto T-2
Test-3	35 Marks	Entire Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Text Books

1. Robert C. Martin, Micah Martin, Agile Software Development: Principles, Patterns, and Practices, Pearson Education, 2006.

References

1. Jeff Sutherland, Scrum: The Art of Doing Twice the Work in Half the Time, Crown Business, 2014.
2. Jeff Patton, User Story Mapping: Discover the Whole Story, Build the Right Product, O'Reilly Media, 2014.
3. Gary McGraw, Software Security: Building Security In, Addison-Wesley, 2006.
4. Dafydd Stuttard, Marcus Pinto, The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Wiley, 2011.
5. Robert C. Seacord, Secure Coding in C and C++, Addison-Wesley Professional, 2013.
6. Adam Shostack, Threat Modeling: Designing for Security, Wiley, 2014.
7. Jim Bird, DevSecOps: A leader's guide to producing secure software without compromising flow, feedback and continuous improvement, O'Reilly Media, 2020.
8. O'Reilly Media, Continuous Security: Reliable Software Releases through Build, Test, and Deployment Automation, O'Reilly, 2020.
9. OWASP Foundation, OWASP Testing Guide, Version 4.2, 2014. (Available online at: https://owasp.org)

Title: Statistical Methods and Data Analysis **Code: CS323**
L-T-P scheme: 3-0-0 **Credit: 3**

Prerequisite: None

Objective:

The objective of course is to equip the students with the mathematical & statistical techniques & their application to business problems. The emphasis will be on the concepts & application rather than derivations. The intention of the course is to make students able to use statistics as a helpful tool for solving complex business research problems under uncertainty and understand methods that quantify issues and give business managers a better basis for making decisions.

Learning Outcomes:

Course Outcome	Description
CO1	To familiarize the concept of data and data categorization and introduce the field of statistics & data analysis.
CO2	To understand and compute various measures of descriptive statistics such as mean, median, standard deviation, skewness, and kurtosis.
CO3	To describe basic concepts of probability and probability distributions and its applications in solving various business problems.
CO4	To learn and apply various statistical techniques such as sampling distribution, interval estimation and hypothesis testing for inferential data analysis using real world examples.
CO5	To develop the understanding to analyze a set of data / real world situations using correlation, regression analysis and ANOVA.
CO6	To build up decision making skills pertinent to the practice of statistics, including the students' abilities to formulate problems, to think creatively, and to synthesize information.

Course Content:

Unit-1: Introduction to Statistics, Types, Scope; Data sources, Data presentation, tabulation, charting, graphs; Measures of central tendency – Mean, Median, Mode; Measures of variations – range, interquartile range, standard deviation; Skewness, moments & kurtosis; Covariance and correlation.
Unit-2: Introduction to probability, basic laws & concepts, conditional probability; Probability distributions, random variable, probability function, expected value and variance, Discrete probability distribution, Binomial Distribution, Poisson Distribution, Continuous Probability Distribution, Normal Distribution, Exponential Distribution.

Unit-3: Sampling – introduction, purpose, random sampling methods, non-random sampling methods; Sampling distributions, Sampling Distribution of the Mean, Central Limit Theorem, Sampling Distribution of the Proportion.
Unit-4: Statistical estimation – Introduction, Properties of a good estimator, Point Estimation, Interval Estimation for sample mean, Interval Estimation for sample proportion, Sample size determination, Hypothesis testing – basic concepts, Null and the Alternative Hypothesis, Tests of Hypotheses about Population Means, Tests of Hypotheses about Population Proportions.

Unit-5: Regression analysis, linear regression, regression lines, regression coefficients, coefficient of determination, Analysis of Variance – introduction, assumptions, computation; One-way classification – variance between samples, variance within samples, F-ratio calculation, Two-way classification.

Teaching Methodology:

The course is a mix of classroom teaching (power point slides) which includes case studies, quiz, problem solving, and numerical questions.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Based on Unit-1 & Unit-2
Test-2	25 Marks	Based on Unit-3 & Unit-4 and around 30% from coverage of Test-1
Test-3	35 Marks	Based on Unit-5 and around 30% from coverage of Test-2
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Learning Resources:

Lectures, tutorials and e-books on Statistical Methods and Data Analysis (are added from time to time): Digital copy will be available on the JUET server.

Text Book:

[1] Anderson, Statistics for Business & Economics, Thomson Learning, Bombay.

Reference Books/Material:

[1] Gupta S.P. & Gupta M.P., Business Statistics, Sultan Chand & Sons, Delhi.
[2] Levin & Rubin, Statistics for Management, Prentice Hall of India, New Delhi.
[3] Mann P., Introductory Statistics, Wiley.
[4] Schumuller J., Statistical Analysis with Excel for Dummies, John Wiley & Sons, NJ.
[5] Berk & Karey, Data Analysis with Microsoft Excel, Cengage Learning, Boston.

Course Title: Computer Graphics **Course Code: 18B14CI643**

L-T Schème : 3-0 **Course Crédits: 3**

Scope and Objectives:

1. To learn and understand the basics of computer graphics applications and graphics devices
2. To learn and understand the geometric figure drawing algorithm on graphic device
3. To learn and understand the Two-Dimensional transformations
4. To learn and understand the Three-Dimensional transformations
5. To understand the concepts of solid modelling and representation
6. To learn about the Visible-Surface, Illumination and Shading

Learning Outcome:

Course Outcome	Description
CO 1	Student will learn about the overview of computer graphic applications and graphics devices (Display Technologies, Raster Refresh (Raster-Scan), CRT, LCD displays, etc.)
CO 2	Student will learn about the scan conversion - lines, circles and Ellipses, filling, clipping and aliasing
CO 3	Student will learn about the Two-Dimensional transformations and matrix representation of 2D Transformations (Translations, Rotation, Reflection, Scaling and Combined Transformation) and Window-to-Viewport transformations
CO 4	Student will learn about the Three-Dimensional transformations and viewing in 3D
CO 5	Student will learn about the solid modelling: representing solids, regularized Boolean Set operations, primitive instancing, sweep representations, spatial-partitioning representations - Octree representation, B-Reps and Constructive Solid Geometry
CO 6	Student will learn about the visible surface detection, illumination and shading

Course Contents:

Unit 1: Introduction to Computer Graphics: Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Input Devices for Operator Interaction, Active and Passive Graphics Devices, Display Technologies, Storage Tube Graphics Displays, Calligraphic Refresh Graphics Displays, Raster Refresh (Raster-Scan) Graphics Displays, Cathode Ray Tube Basics, Colour CRT Raster Scan Basics, Video Basics, The Video Controller, Random-Scan Display Processor, LCD displays, Touch screen, Graphics Primitives.

Unit II: Scan conversion – lines, circles and Ellipses; Filling polygons and clipping algorithms, Scan Converting Lines, Mid-point criteria, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Scan Converting Ellipses, Filling Polygons, edge data structure, Clipping Lines algorithms– Cyrus-Beck, Cohen-Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components

Unit III: Two-Dimensional Transformations: Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The Unit Square, Solid Body Transformations, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, The Window-to-Viewport Transformations.

Unit IV : Three-Dimensional Transformations and Viewing in 3D: Introduction, Three-Dimensional Scaling, Three-Dimensional Shearing, Three-Dimensional Rotation, Three-Dimensional Reflection, Three-Dimensional Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations, Affine and Perspective Geometry, Perspective Transformations, Techniques for Generating Perspective Views, Vanishing Points, the Perspective Geometry and camera models, Orthographic Projections, Axonometric Projections, Oblique Projections, View volumes for projections.

Unit V: Solid Modelling: Representing Solids, Regularized Boolean Set Operations, Primitive Instancing, Sweep Representations, Spatial-Partitioning Representations: Octree representation, B-Reps, Constructive Solid Geometry, Comparison of Representations

Unit VI: Visible-Surface Determination: Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, The z-Buffer Algorithm, Scan-line method, Painter's algorithms (depth sorting), Area sub-division method, BSP trees, Visible-Surface Ray Tracing, comparison of the methods

Unit VII: Illumination and Shading: Illumination and Shading Models for Polygons, Reflectance properties of surfaces, Ambient, Specular and Diffuse reflections, Atmospheric attenuation, Phong's model, Gouraud shading, some examples.

Unit VIII: Image Manipulation and Storage: What is an Image? Digital image file formats, Image compression standard – JPEG, Image Processing - Digital image enhancement, contrast stretching, Histogram Equalization, smoothing and median Filtering.

Evaluation Scheme:

Exams	**Marks**	**Coverage**
Test-1	15 Marks	Syllabus covered upto T-1
Test-2	25 Marks	Syllabus covered upto T-2
Test-3	35 Marks	Entire Syllabus
Assignment	10 Marks	
Tutorials	5 Marks	
Quiz	5 Marks	
Attendance	5 Marks	
Total	**100 Marks**	

Text book:

1. D. Hearn and M. P. Baker, Computer Graphics using C

Reference Books:

1. Foley, Van Darn, Feiner, Hughes, Computer Graphics, Second edition
2. D.F. Rogers, Mathematical elements for computer graphics, Second edition
3. Rogers, Procedural elements for Computer Graphics, Second edition

Online Resources:

1. https://nptel.ac.in/courses/106106090
2. https://in.coursera.org/learn/interactive-computer-graphics