Assam University, Silchar

DEPARTMENT OF COMPUTER SCIENCE

Four Year Undergraduate (FYUG) leading to M. Sc. (Integrated) 5-year Programme

Under NEP 2020

w.e.f. 2023-24

Department of Computer Science
Assam University, Silchar

Curriculum for Four Year Undergraduate (FYUG)/M. Sc. (Integrated) 5-year Programme in Computer Science under NEP 2020 w.e.f. 2023-24

The Department of Computer Science offers B. Sc. (Computer Science) (Honours with Research), a Four Year Undergraduate Degree Programme (FYUG), M. Sc. (Integrated) 5-year programme and M. Sc. Degrees in Computer Science under the provisions of NEP 2020 effective from the Academic Year 2023-24.

Eligibility Criteria for admission to Four Year Undergraduate Degree programme/M. Sc. (Integrated) 5-year Programme in Computer Science:
Candidates who have passed the HS examination (10+2) in Science stream with Physics and Mathematics (in combination with any other science subjects) with at least 50% marks in aggregate are eligible for admission into the first semester of M. Sc. (Integrated) 5-year programme. Admission will be based on the scores of CUET conducted by National Testing Agency (NTA), Govt. of India.

Eligibility for promotion of students with three years undergraduate degree to the 7th Semester of the M. Sc. (Integrated) 5-year Programme (That is in the first semester of the PG level of the 5 year M. Sc. Integrated Programme), a student has to pass in all the papers up to the 6th semester of the Four Year Undergraduate Degree Programme and has to successfully complete the undergraduate degree of three years duration.

Eligibility for promotion of students with four years undergraduate degree to the 9th Semester of the M. Sc. (Integrated) 5-year Programme (That is in the third semester of the PG level of the M. Sc. (Integrated) 5-year Programme), a student has to pass in all the papers up to the 8th semester of the Four Year Undergraduate Degree Programme and has to successfully complete the undergraduate degree (Honours with Research) under the provisions of NEP 2020.

Course Mapping for CUET: 1. English 2. Three domain specific subjects *viz*. Physics and Mathematics as compulsory subjects and any one other science subject.

Selection of candidates for admission to the Four Year Undergraduate (FYUG)/M.Sc. (Integrated) 5-year Programme in Computer Science will be made on the basis of NTA-CUET score only.

Programme Specific Outcome

On successful completion of the FYUG Computer Science programme, students will be able to:

1. Understand and apply the concepts of computer architecture, data structure, networking and operating system.
2. Write and execute programs in various programming languages to solve real life problems.
3. Learn and apply techniques of database management systems and design data bases for various applications.
4. Understand the theories and algorithms of Artificial Intelligence including Machine Learning and Data Science and apply these to solve problems.
5. Perform analysis of algorithms related to space and time complexity of Algorithms.
6. Learn and apply the knowledge of Internet Technologies and Cyber Security.
7. Develop theoretical as well as practical knowledge of Software development.
8. Seek admission in Post-graduate/Research Programmes in Computer Science, Information Technology and Computer Application etc.
9. Seek employment in various jobs in the Government sector as well as IT and related industries and perform various roles related to software development, testing and maintenance.

SEMESTER-I

Course Code	Type of Course	Course Name	L	T	P	Credit	Sessional Marks	End Semester Marks	Total Marks
CS-DSC-101	Discipline Specific Core	Programming in C	3	-	-	3	30	70	100
CS-DSC-102	Discipline Specific Core	Digital Logic and Switching Theory	3	-	-	3	30	70	100
CS-DSM-101	Discipline Specific Minor	Mathematics-I	3	-	-	3	30	70	100
CS-IDC-101	Interdisciplinary Course	Fundamentals of Computer and Applications	3	-	-	3	30	70	100
CS-AEC-101 MIL-101	Ability Enhancement Courses (Language)	MIL-I	2	-	-	2	-	50	50
CS-SEC-101	Skill Enhancement Course	Lab on a) Programming in C b) Digital logic and Switching Theory	-	-	6	3	30	70	100
CS-VAC-101	Common Value Added Courses	NCC/NSS/Sports/ HW/UI/GCS/Yoga				3	30	70	100
									SEMESTER I Total Marks: 650

SEMESTER-II

Course Code	Type of Course	Course Name	L	T	P	C	Sessional Marks	End Semester Marks	Total Marks
CS-DSC-151	Discipline Specific Core	Python Programming	3	-	-	3	30	70	100
CS-DSC-152	Discipline Specific Core	Numerical Methods	3	-	-	3	30	70	100
CS-DSM-151	Discipline Specific Minor	Mathematics-II	3	-	-	3	30	70	100
CS-IDC-151	Interdisciplinary Course	Introduction to Internet Technology	3	-	-	3	30	70	100
CS-AEC-151 EL-151	Ability Enhancement Courses (Language)	English-I	2	-	-	2	-	50	50
CS-SEC-151	Skill Enhancement Course	Lab on a) Python Programming b) Numerical Methods	-	-	6	3	30	70	100
CS-VAC-151	Common Value Added Courses	EVS				3	30	70	100
Undergraduate Certificate: 40 Credits									**SEMESTER II Total Marks: 650**

SEMESTER-III

Course Code	Type of Course	Course Name	L	T	P	C	Sessional Marks	End Semester Marks	Total Marks
CS-DSC-201	Discipline Specific Core	Data Structure	3	1	-	4	30	70	100
CS-DSC-202	Discipline Specific Core	Computer Architecture	3	1	-	4	30	70	100
CS-DSM-201	Discipline Specific Minor	Introduction to Probability and Statistics	3	1	-	4	30	70	100
CS-IDC-201	Interdisciplinary Course	Cyber Security	3	-	-	3	30	70	100
CS-AEC-201 MIL-201	Ability Enhancement Courses (Language)	MIL-II	2	-	-	2	-	50	50
CS-SEC-201	Skill Enhancement Course	Lab on Data Structure	-	-	6	3	30	70	100
SEMESTER III Total Marks: 550									

SEMESTER IV

Course Code	Type of Course	Course Name	L	T	P	C	Sessional Marks	End Semester Marks	Total Marks
CS-DSC-251	Discipline Specific Core	Database Management System	3	1	-	4	30	70	100
CS-DSC-252	Discipline Specific Core	Microprocessor	3	1	-	4	30	70	100
CS-DSC-253	Discipline Specific Core	Discrete Mathematics	3	1	-	4	30	70	100
CS-DSM-251	Discipline Specific Minor	Lab on a) Database Management System b) Microprocessor	-	-	6	3	30	70	100
CS-DSM-252	Discipline Specific Minor	Data Communication and Computer Networks	3	-	-	3	30	70	100
CS-AEC-251 EL-251	Ability Enhancement Courses (Language)	English-II	2	-	-	2	-	50	50
Undergraduate Diploma: 80 Credits									**SEMESTER IV Total Marks: 550**

SEMESTER V

Course Code	Type of Course	Course Name	L	T	P	C	Sessional Marks	End Semester Marks	Total Marks
CS-DSC-301	Discipline Specific Core	Operating System	3	1	-	4	30	70	100
CS-DSC-302	Discipline Specific Core	System Software	3	1	-	4	30	70	100
CS-DSC-303	Discipline Specific Core	Computer Graphics	3	1	-	4	30	70	100
CS-DSM-301	Discipline Specific Minor	Lab on a) Operating System b) Computer Graphics	-	-	6	3	30	70	100
CS-DSM-302	Discipline Specific Minor	Simulation and Modeling	3	-	-	3	30	70	100
CS-SEC-301		Internship/ Community Engagement/ Field Study				2	15	35	50
									SEMESTER V Total Marks: 550

SEMESTER VI

Course Code	Type of Course	Course Name	L	T	P	C	Sessional Marks	End Semester Marks	Total Marks
CS-DSC-351	Discipline Specific Core	Object Oriented Programming with C++	3	1	-	4	30	70	100
CS-DSC-352	Discipline Specific Core	Programming in JAVA	3	1	-	4	30	70	100
CS-DSC-353	Discipline Specific Core	Wireless and Mobile Computing	3	1	-	4	30	70	100
CS-DSC-354	Discipline Specific Core	System Analysis and Design	3	1	-	4	30	70	100
CS-DSM-351	Discipline Specific Minor	Lab on a) Object Oriented Programming with C++ b) Programming in JAVA	-	-	8	4	30	70	100
Bachelor's Degree: 120 credits		**Min 7.5 CGPA to Move to 4th year**					**SEMESTER VI Total Marks: 500**		

SEMESTER VII

Course Code	Type of Course	Course Name	L	T	P	C	Sessional Marks	End Semester Marks	Total Marks
CS-DSC-401	Discipline Specific Core	Design and Analysis of Computer Algorithms	3	1	-	4	30	70	100
CS-DSC-402	Discipline Specific Core	Theory of Computation	3	1	-	4	30	70	100
CS-DSC-403	Discipline Specific Core	Artificial Intelligence	3	1	-	4	30	70	100
CS-DSC-404	Discipline Specific Core	Machine Learning	3	1	-	4	30	70	100
CS-DSM-401	Discipline Specific Minor	Lab on a) Design and Analysis of Computer Algorithms b) Artificial Intelligence	-	-	8	4	30	70	100
SEMESTER-VII Total Marks: 500									

SEMESTER VIII

Course Code	Type of Course	Course Name	L	T	P	C	Sessional Marks	End Semester Marks	Total Marks
CS-DSC-451	Discipline Specific Core	Principles of Compiler Design	3	1	-	4	30	70	100
CS-DSM-451	Discipline Specific Minor	Data Science and Research Methodology	3	1	-	4	30	70	100
CS-DSC-452		Research Project/ Dissertation	-	-	24	12	90	210	300
Bachelor's Degree (Honours with Research): 160 Credits							**SEMESTER VIII Total Mark: 500**		

Semester I

Course Code: CS-DSC-101 **Course Title: Programming in C**

Credits: 3 **LTP: 3 – 0 – 0**

Course objectives:
1. To introduce the concepts of programming and programming language C.
2. To explain the concepts of functions and programme structure in C
3. To explain how to write and implement C programs
4. To explain the concept and working of pointers and files in C
5. To introduce the low level programming in C

***Course outcomes*:**
After successful completion of the course, the students will be able to:
1. Apply the concepts of Programming in C
2. Apply thoroughly the building blocks of C programming language
3. Apply and implement C programs and solve problems through programming
4. Apply the Concept of pointers and files in C & Programming with C.
5. Design and implement programs using pointers and files in C.

UNIT-I:
Fundamentals of computer programming with C – Data Types, Expressions, Operations – input, output; Writing simple C programs; Control structures (WHILE, DO-WHILE, FOR, IF-ELSE, SWITCH, BREAK, CONTINUE, GOTO STATEMENTS, nested loops etc.) and writing programs using control structures; solving elementary programming problems from various areas of applications including mathematics and statistics.

UNIT-II:
Functions and program structure – Defining and accessing functions in C, passing arguments to a function, specifying argument data types – Illustration with example programs and problem solving through programs; Function prototypes, Functions returning non integers; Storage classes – Automatic, External, Static and Register variables, Scope rules, Header files, Block structure; Recursion in C – writing recursive programs and problem solving, The C Preprocessor

UNIT-III:
Definition and array processing, passing arrays to a function, multidimensional arrays, Arrays and Strings; POINTERS – pointers and addresses, pointer declaration, pointers and

function arguments – passing pointers to a function, Pointers and one dimensional arrays; Address arithmetic – operations on pointers, character pointers and functions; Pointer arrays/arrays of pointers, pointers to pointers, initialization of pointer arrays, pointers and multidimensional arrays; Command line arguments, Pointers to functions, passing functions to other functions.

UNIT-IV:

Structures and Unions – Basics of structures, processing of structures, user defined data types (typedef), Structures and Pointers, Structures and functions – passing structures to a function, Arrays of structures, Pointers to structures, Self-referential structures, Table lookup, UNIONS. writing programs and problem solving with structure and union

UNIT-V:

Input and output – Standard input and output, Formatted output – printf, Variable length argument, Formatted input - scanf; Data files – opening and closing data file, creating a data file, processing a data file, file access, unformatted data files, miscellaneous function in C; Low Level programming – Register variables, Bitwise operations, Bit fields, Enumeration, Commands Line arguments/parameters, Library functions, Macros, The C preprocessor.

Text Books:

1. The C Programming Language – Brian W. Kernighan and Denis M. Ritchie (PHI), Latest Edition
2. Theory and Problems of Programming with C – Byron S. Gottfried, (McGraw Hill), Latest Edition.
3. Programming with C - E. Balaguruswamy, McGraw Hill (Latest Edition)
4. Programming with C - Rajaraman R., PHI (Latest Edition)

Reference Books:

1. Let Us C – Yashavant P. Kanetkar, BPB Publications (Latest Edition)
2. Pointers in C – Yashavant P. Kanetkar, BPB Publications (Latest Edition)
3. Programming with ANSI C - B.T. Holmes, BPB (Latest Edition)

Course Code: CS-DSC-102 **Course Title: Digital Logic & Switching theory**

Credits: 3 **LTP: 3 – 0 – 0**

Course Objectives:

1. To explain the basics of digital logic and the concept of current flow in circuits.
2. To explain the different number systems.
3. To explain the binary, octal and hexadecimal operations.
4. To explain the design of the digital circuits using different gates.
5. To explain the combinational and sequential circuits.

Course Outcomes:

Students who complete the course will be able to do the following:

1. Convert any number from one number system to another.
2. Design any digital circuit.
3. Perform basic operations of gates.
4. Conceive the idea of memory in circuits.
5. Extend the digital logic to PLA, ROM etc.

UNIT-I:

Number Systems & Codes: Philosophy of number systems – complement representation of negative numbers, binary arithmetic- addition, subtraction, multiplication ,division, binary codes conversion-binary to decimal, binary to hexadecimal. Binary to octal, octal to binary, hexadecimal to binary, floating point representation.

UNIT-II:

Boolean Algebra and Switching Functions: Fundamental postulates of Boolean Algebra - Basic theorems and properties - switching functions–Canonical and Standard forms-Algebraic simplification digital logic gates, properties of XOR gates –universal gates-Multilevel NAND/NOR realizations.

Minimization of Switching Functions: Map method, Prime implicants, Don't care combinations, Minimal SOP and POS forms, Tabular Method, Prime –Implicant chart, simplification rules.

UNIT-III:

Combinational Logic Design: Design using conventional logic gates, Encoder, Decoder, Adders, Substractors Multiplexer, Demultiplexer, Modular design using IC chips, MUX Realization of switching functions Parity bit generator, Code-converters, Hazards.

UNIT-IV:

Sequential Circuits -I: Classification of sequential circuits (Synchronous, Asynchronous, Pulse mode, Level mode with examples) Basic flip-flops-Triggering and excitation tables. Steps in synchronous sequential circuit design. Design of modulo-N Ring & Shift counters, Serial binary adder, sequence detector.

UNIT-V:

Programmable Logic Devices, Threshold Logic: Basic PLD's-ROM, PROM, PLA, PLD Realization of Switching functions using PLD's. Capabilities and limitations of Threshold gate, Synthesis of Threshold functions, Multigate Synthesis.

TEXT BOOKS

1. Digital Logic and Computer Design- M. Morris Mano – Pearson Education, 2008.
2. Digital Design – Morris Mano, PHI, 3rd Edition, 2006.

REFERENCE BOOKS

1. An Engineering Approach to Digital Design – Fletcher, PHI, 2010.
2. Malvino A.P, Digital Principles and Applications, Tata McGraw Hill, 2009.

E – BOOKS

1. https://ebooks.lpude.in/computer_application/ad/DCAP108_DIGITAL_CIRCUITS_AND_LOGIC_DESIGNS.pdf
2. https://www.academia.edu/37310765/Digital_Logic_Design_4th_edition

MOOCs

1. https://nptel.ac.in/courses/117105080

2. https://www.udemy.com/course/digital-electronics-logic-design

Course Code: CS-DSM-101	**Course Title: Mathematics – I**
Credits: 3	**LTP: 3 – 0 – 0**

Course Objectives:

This course is designed to provide a deeper and rigorous understanding of fundamental concepts viz Set theory and functions, Matrix theory, Vector analysis, Differential Equations, Laplace transforms, Fourier Transforms and Integral Transforms etc. The main focus of this course will be on the theoretical foundation of the above said concepts and it will cultivate the rigorous Mathematical skills in the students.

Course Outcomes:

At the end of the course, the students will be able to:

1. Apply the knowledge of concepts of set theory and functions in order to further study and explore the concepts of Differential equations and other Mathematical concepts.
2. Apply Ordinary Differential Equations of various types, their solutions and fundamental concepts about their existence.
3. Apply various concepts of Matrix theory, Eigenvalues, Eigen vectors and their applications in Linear Algebra and Vector algebra.
4. Apply the concepts of Laplace Transforms and Fourier Transforms.

UNIT-I:

Set theory and Functions: set, subsets, union of sets, intersection of sets, difference of two sets, symmetric difference of two sets, van diagram, De-Morgan laws, distributive property, Cartesian product, function, one-one function, onto function, bijective function, composition of functions and inverse function.

UNIT-II:

Matrix Theory: Symmetric and skew symmetric matrices, Hermitian and skew-Hermitian matrices, minor and cofactors, orthogonal and singular matrix, adjoint and inverse of matrices, application of matrices for solving linear system of equations, rank of a matrix, Eigenvalues, Eigen vectors, characteristic equation of a matrix, Cayley-Hamilton theorem and its use for finding inverse of matrices.

UNIT-III:

Vector Analysis: Introduction of vectors, Vector equation of straight lines, plane, circle and spheres, Scalar product of two vectors, Vector product of two vectors, scalar triple product, vector triple product, directional derivative, gradient of a scalar (function), divergence and curl of vectors, physical interpretation of gradient, divergence & curl, linear dependency and independency of vectors.

UNIT-IV:

Differential Equations: order and degree of differential equation, linear equations, Solution of first order linear differential equations by variable separable method, Homogeneous equations, linear equations, exact equations, Higher order linear differential equations with constant coefficients, Method of undetermined coefficients and Variation of parameters.

UNIT-V:

Integral transforms: Laplace transforms, Fourier series and Fourier transforms, related problems.

Text Books/ References:

1. Higher Algebra: Abstract & Linear- S.K. Mapa, Sarat Book House, 2003. (Unit I & II).
2. Vector Algebra: Shanti Narayan & PK Mittal, S. Chand & Co. Ltd.,2005. (Unit III).
3. Ordinary Differential Equations With Applications and Historical Notes: G.F. Simmons, Tata McGraw Hill, Second Edition. (Unit IV & V).

Course Code: CS-IDC-101 Course Title: Fundamentals of Computer & applications

Credits: 3 LTP: 3 – 0 – 0

Course objectives:

1. To introduce the concepts of Computer System to the students of UG level having no prior knowledge of
   Computer
2. To explain the concepts of basic units of Computer and its organization..

3. To explain data representation techniques and computer Software systems.
4. To explain working with windows operating system and MS Office

***Course Outcome*:**
After successful completion of the course, the students will be able to:
1. Apply the concepts of Computer System, its functions and working
2. Apply the basic units of Computer system and its organization
3. Apply about operating systems and its working
4. Work with windows Operating System and various software packages of MS Office

UNIT 1:
Introduction to Computer System: Introduction to Computer System: Definition of Computer, and Basic Characteristics and Operations performed by Computers, History and Generations of Computers, Classification of Computers.

Basic Organization of Computer Systems: Fundamental model (Von-Neuman) and Block diagram of Computer: ALU, CU, CPU; Input, Output and Storage Units and their Functions; Types of storages and storage devices used in Computer Systems, Input and Output Devices.

Introduction to Number Systems – Decimal, Binary, Octal and Hexadecimal number systems and conversion from one system to another

UNIT II:
Data Representation and Computer Software Systems: Data Representation**:** Representation of data in Binary and data types: Numeric, Alphabetic and Alphanumeric; Computer codes: Most commonly used computer codes – Bits and Bytes; BCD, EBCDIC, ASCII and UNICODE; Introductory concept of Boolean Algebra and Basic Gates: AND, OR, NOT, NAND, XOR.

Computer Software Systems – A mechanism for Human Computer Interface; Concept of Software and Types of Software: System Software, Application Software, Firmware, Middleware. Software Development Steps. Computer Languages and its Types: Machine Language, Assembly Language, High Level Languages: advantages and disadvantages of computer languages. Concept of Translators: Compiler, Linker, Loader, Interpreter and Operating System.Concept of Computer Programs Design and Development: Algorithms, Flowchart, Pseudocodes, Control Structures, Illustrative Examples

UNIT III:
Operating Systems, Introduction to Operating Systems: Types of Operating Systems: Uniprocessor and Multiprocessor Single user and Multiuser, Overview of Functions of Operating System – Process management, Memory management, File management, Processor and Device Management, Security and Command Interpretation, Overview of UNIX and LINUX Systems. Introduction to computer networking

Windows Operating System: Introduction to Windows, Starting Windows, Desk Top, Task Bar, Start Up Menu Working with programs and icons-Adding, removing, starting and quitting programs and icons. Working with files and folders-creating, deleting, opening, finding, copying, moving and renaming files and folders. Control Panel, setting, My Computer, Recycle bin, My documents, drives. Windows notepad, Accessories and Windows Explorer.

UNIT IV:

Word Processor and SpreadSheet, MS-Word: Overview of Word Processing, Word Processor and its features, Parts of word window, Types of Menus, Opening, creating saving, cut, copy and paste, save & save as. Editing of Text, Find and Replace, print and print preview. Word count, Bullets and Numbering, Spell Checker, Grammar Checker, Auto Correct, Auto Complete, Auto Text, Header and footer, tables, mail merge, border and shading, page setup, printing. Mail Merge, Table handling and important shortcut keys, Macros.

Spreadsheet: MS Excel and its features, Entering Information in Worksheet, Editing Cell Entry, Moving and Copying Data, deleting or Inserting Cells, Rows and Columns, Custom Numeric Formats, Using Formulas and functions, Creating charts, Sorting and Searching.

UNIT V:

Presentation Software and Usage , MS-PowerPoint: Presentation Softwares and its uses, Overview of MS-PowerPoint, Steps for creating PowerPoint Presentation, Slides, PowerPoint Views, Auto content, Assigning Slide Transitions, Using Preset Animations, Hiding Slides, Slide Show, Controlling the Slide Show with a Keyboard, Setting Slide Show Timings, Wizard, Custom Animation, Transition and build effects, Printing slides and important shortcut.

Use of Computers in Education and Research: Data analysis, e-Library, Google Scholar

Text Books:

1. P.K Sinha & Priti Sinha, Computer Fundamentals, BPB Publications, (Latest Edition)
2. V. Rajaraman, Fundamentals of Computers, 6 th edition PHI Learning Private Limited 2014 (Latest Edition)
3. R.K. Taxali, PC Software for Windows (Latest Edition)

Reference Books:

1. Alexis Leon and Matthews Leon: Introduction to Computers, Leon Vikas,1999.
2. Suresh K. Basandra, Computer Systems Today, Galgotia Publications.
3. Peter Norton: Computing Fundamentals. 6 th Edition, McGraw Hill-Osborne,2007
4. Joyce Coax , Joan Preppernau,,Steve Lambert and Curtis Frye,2007
5. Microsoft Office System step by step, Microsoft Press (Latest Edition)
6. Microsoft Office 20xx Training Guide, BPB Publications (Latest Edition)

Course Code: CS-AEC-101 MIL-101 **Course Title: MIL-I**

Credits: 2 **LTP: 2 – 0 – 0**

Course Code: CS-SEC-101 **Course Title: a) Lab on programming in C**

Credits: 3 **LTP: 0 – 0 – 6**

Course Objectives:
1. To explain design and implementation of C programs
2. To explain writing and executing programs with control structures and functions
3. To explain writing and executing programs with pointers in C
4. To explain writing and executing programs with structures and unions
5. To explain writing and executing programs with files in C

Course outcomes:
After successful completion of the course students will be able to:
1. Write programs and implement programs in C.
2.Write programs and execute programs with control structures and functions
3. Write and execute programs with pointers in C
3. Write and execute programs with structures and unions
4. Write and execute programs with files in C

Problem solving of various nature by implementing programs in C Programming languages based on unit wise contents of the theory paper Programming with C. Following are some programming tasks for laboratory programming assignments but the assignments are not limited to these only.

List of laboratory programming assignments (not limited to these):

1. Write a program to

a) Produce ASCII equivalent of given number

b) Find the divisor or factorial of a given number.

c) Evaluate the following algebraic expressions after reading necessary values from

the user (ax+b)/ (ax-b) -2.5 log x-cos 30+|x2-y2|+sqrt (2xy) - (x5+10x4+8 x 3+4x+2)

d) Find sum of a geometric series

e) Cipher a string

f) Check whether a given string follows English capitalization rules

g) Find sum of the following series 1+ ½ + 1/3 +________+1/20

h) Search whether a given substring exist in an input string or not and then delete this string from input string.

2. Write a recursive program for tower of Hanoi problem

3. The Fibonacci sequence of numbers is 1, 1, 2, 3, 5, 8……. Based on the recurrence relation F(*n*)=F(*n*-1)+F(*n*-2) for n>2 Write a recursive program to print the first *n* Fibonacci number

4. Write a menu driven program for matrices to do the following operation depending on whether the operation requires one or two matrices

a) Addition of two matrices

b) Subtraction of two matrices

c) Finding upper and lower triangular matrices

d) Trace of a matrix

e) Transpose of a matrix

f) Check of matrix symmetry

g) Product of two matrices.

5. Write a program that takes two operands and one operator from the user perform the operation and then print the answer

7. Write functions to add, subtract, multiply and divide two complex numbers (x+iy) and (a+ib) Also write the main program.

8. Write a menu driven program for searching and sorting with following options:-

a) Searching (1) Linear searching (2) Binary searching

b) Sorting (1) Insertion sort (2) Selection sort

9. Write a program to copy one file to another, use command line arguments.

10. Write a program to mask some bit of a number (using bit operations)

11. An array of records contains information of managers and workers of a company. Print all the data of managers and workers in separate files.

Course Code: CS-SEC-101 Course Title: b) Lab on Digital Logic & switching theory

Course Objectives:

1. To demonstrate the basics of digital logic and concept of current flow in circuits.
2. To demonstrate the different number (binary, oct, hex) conversion.
3. To demonstrate the basic truth tables and gate operations.
4. To demonstrate the combinational circuits.
5. To demonstrate the sequential circuits.

Course Outcomes:

After successful completion of the course, the students will be able to:
1. Convert any number from one number system to another.
2. Design any digital circuit.
3. Perform basic operations of gates.
4. Conceive the idea of memory in circuits.
5. Extend the digital logic to PLA, ROM etc.

Following are some samples for laboratory assignments but the assignments are not limited to these only:

01: To study the digital board DB-01 and to verify the truth tables for AND, OR, NOT, NAND, NOR and XOR gates.

02: To study the digital board DB-02 and to verify the truth tables for Universal gates (NAND, NOR).

03: To study the digital board DB-03 and to implement EX-OR gate.

04: To study the digital board DB-04 and analyze De-Morgan's Theorem.

05: To study the digital board DB-05 and to study the Ex-OR gate implementation for
   a. Odd parity generator
   b. Even parity generator
   c. Binary word comparator

06: To study the digital board DB-06 and to study the code conversion circuits for
   a. Binary to gray code
   b. Gray to binary code

07: To study the digital board DB-07 and to verify BCD to Excess 3 code conversion circuit.

08: To study the digital board DB-09 and to verify the truth tables for
   a. 8 to 3 line encoder
   b. 3 to 8 line decoder

09: To study the digital board DB-10 and to verify the truth tables for Multiplexer(MUX) &DeMultiplexer (DMUX).

10: To study the digital board DB-11 and to verify the truth tables for the R-S flip flops, D flip flops, J-K flip flops & T flip flops.

TEXT BOOKS

1. Digital Logic and Computer Organisation- V. Rajaraman & T. Radhakrishnan–PHI, 2006.
2. Digital Logic Fundamentals – Ananthi S. and J.G. Sheshasaayee, Margham Publications, 2010.

REFERENCE BOOKS

1. An Engineering Approach to Digital Design – Fletcher, PHI, 2010.
2. Malvino A.P, Digital Principles and Applications, Tata McGraw Hill, 2009.

E – BOOKS

1. https://ebooks.lpude.in/computer_application/ad/DCAP108_DIGITAL_CIRCUITS_AND_LOGIC_DESIGNS.pdf
2. https://www.academia.edu/37310765/Digital_Logic_Design_4th_edition

MOOCs

1. https://nptel.ac.in/courses/117105080

2. https://www.udemy.com/course/digital-electronics-logic-design

Course Code: CS-VAC-101 **Course Title: NCC/NSS/Sports/HW/UI/GCS/Yoga**

Credits: 3

Semester II

Course Code: CS-DSC-151 **Course Title: Python Programming**

CREDIT: 3 **L-T-P: 3-0-0**

Course Objectives:
To understand why Python is a useful scripting language for developers; to learn how to use lists, tuples, and dictionaries, sets in Python programs; to learn how to use indexing and slicing to access data in Python programs ; to learn how to write loops and decision statements in Python; to learn how to write functions and pass arguments in Python; to learn how to build and package Python modules for reusability; to learn how to read and write files in Python; to learn how to design object-oriented programs with Python classes ; to learn how to use class inheritance in Python for reusability; to learn how to use exception handling in Python applications for error handling ; to learn how to use Machine Learning tools like Pytorch, keras, TensorFlow etc.

Course Outcomes:
At the end of the course it is expected that a student would be reasonably proficient in writing Python programs for solving various problems as the course covers topics ranging from basics of Python Programming to advanced level.

UNIT-I
Introduction: Basic Elements of Python, Operators, Python Statements & Comments, Python Type Conversion, Indentation in Python, print() and input() functions, Strings, Python IDEs
Python Flow Control: if…else , for loop, while loop, break and continue, Python Pass, range statement; Python memory model: names, mutable and immutable values

UNIT-II
Python Functions: Python Functions, Function Arguments, Recursion, Inductive function definition, Anonymous Function, Lambda function, Passing functions as arguments, Python Global Keyword, Python Modules, Python Package.

UNIT-III
Python Collections: List, Tuple, Sets and Dictionary; String Manipulation: Basic Operations, Slicing, Python Regular expressions; Python iterators, Python Generators, Python Closure, Python Decorators, Higher order functions on lists: map, filter, list comprehension

Unit IV
Python Files: Python File Operation, Python Directory, Python Exception Handling, User defined exception; Assertions; Classes and objects in Python; MATPLOTLIB

Unit V

Arrays vs lists, Scope in Python: local, global, nonlocal names, Nested functions, Binary Search, Data structures: stack, queue implementation in Python, Sorting: Merge sort , Quicksort, Stable sorting implementation in Python, Linked lists: find, insert, delete, Binary search trees: find, insert, delete

Text Books:

1. Introduction to Computation and Programming Using Python, John V. Guttag, PHI
2. Core Python Programming, Dr. R.Nageswara Rao, Dreamtech Press

Reference:

1. Swayam course on "Programming, Data Structures And Algorithms Using Python" By Prof. Madhavan Mukund, Chennai Mathematical Institute

Course Code: CS-DSC-152 **Course Title: Numerical Methods**

CREDIT: 3 **L-T-P: 3-0-0**

Course Objectives:

The objective of this course is to familiarize the various numerical methods to solve scientific problems. This course covers the designing and understanding of iterative algorithms to solve numerical problems.

Course Outcomes:

At the end of this course student will be able to:

1. Apply the different types of errors in numerical methods.
2. Apply the designing of iterative algorithms to solve various numerical problems.
3. Able to apply the basic numerical methods to to solve nonlinear equations, set of linear equations, numerical approximation methods and numerical integration and numerical differentiation.
4. Able to apply the uses and importance of iterative algorithms to solve numerical problems

UNIT-I:

Approximation and Error in Computing**:** Introduction, Significant digits, different types of error, Absolute and relative error, Error estimation, Floating point arithmetic and Round off error. Solution of nonlinear equations: Bisection's method, Newton-Raphson's Method, Secant's Method.

UNIT-II:

Interpolation: Introduction, Errors in Polynomial Interpolation, Finite differences: Forward

Differences- Backward differences, Differences of a polynomial, Newton's formula for interpolation, Central difference interpolation Formula – Gauss Central Difference Formula, Lagrange's Interpolation formula.

UNIT-III:
Solution of linear system: Matrix inversion method, Gauss Elimination, Gauss-jordan method. Curve Fitting: Fitting a straight line, Second degree curve-exponential, curve-power curve by method of least squares.

UNIT-IV:
Numerical Differentiation and Integration: Cubic spline method, Trapezoidal rule – Simpson's 1/3 Rule –Simpson's 3/8 Rule. Partial differential equation: Laplace's equation-Jacobi's Method- Gauss seidel method.

UNIT-V:
Numerical solution of Ordinary Differential equations: Solution by Taylor's series-Picard's Method of successive Approximations-Euler's Method-Runge-Kutta Methods, Predictor-Corrector Methods, Adams- Moulton Method, Milne's Method.

Text Books/References:

1. Balagurusamy, E., "Numerical Methods", Tata McGraw-Hill Pub.Co.Ltd, New Delhi, 1999.
2. D.Kincaid, W.cheney,"numerical Analysis",Brooks/Cole Publishing Company, California,2001.
3. S.S. Sastry, "Introductory Methods of Numerical Analysis", PHI learning Pvt Ltd, 5thEdition 2010.
4. Computer Oriented Numerical Methods - Rajaraman, PHI
5. Numerical Computations - Venkataraman

Course Code: CS-DSM-151 **Course Title: Mathematics-II**

Credit: 3 **LTP: 3-0-0**

Course Objectives :
This course is designed to provide a deeper and rigorous understanding of Real Analysis. The main focus of the course will be on the theoretical foundation of Differential and Integral Calculus.

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

1. Apply the nature of abstract Mathematics and explore the concepts in further details .
2. Identify challenging problems in real variable theory and find their appropriate

solutions .

3. Deal with axiomatic structure of Differential Calculus and generalize the concepts of sequences, series, and continuous functions.

UNIT-I:

Differential Calculus: Limit, Cauchy's criteria for existence of limits (without proof), problems on limit, Continuity: $\varepsilon-\delta$ definition of continuity, problems on continuity, bounded functions, properties of continuous and bounded functions.

UNIT-II:

Differentiability: $\varepsilon-\delta$ definition of derivative, relation between continuity and differentiability, Intermediate value theorem, Rolle's Theorem, mean value theorem, L'Hospital rule and Taylor's theorem.

UNIT-III:

Successive differentiation, Leibnitz theorem, curvature, asymptotes, singular points, Functions of several variables: partial derivative, total differentials, Euler's theorem of homogeneous function of two variables, Jacobian, maxima, minima, necessary and sufficient conditions for maxima and minima.

UNIT-IV:

Sequence: Convergent sequence, monotone sequence, subsequences and Bolzano-Weierstrass theorem, Cauchy criterion for convergence of sequence and divergent sequences.

Series: Convergence and absolute convergence of series, limit comparison test, root test, ratio test, integral test, raabe's test, alternating series test and tests for non absolute convergence of a series.

UNIT-V:

Integral Calculus: definition and properties of definite integrals, Riemann integrable functions, Fundamental theorem, Area bounded by plane curves, Volumes and surfaces of solid of revolution about axis.

Text Books:

1. Differential Calculus: Das & Mukherjee, U.N. Dhur Publishers, 1975. (Unit -I, II, III, IV)
2. Integral Calculus: Das & Mukherjee, U.N. Dhur Publishers, 1998. (Unit – V)

Reference:

1. Introduction To Real Analysis: Bartle & Sherbert, Wiley Student Edition, Third edition.

Course Code: CS-IDC-151 **Course Title: Introduction to Internet Technology**

CREDIT: 3 **L-T-P: 3-0-0**

Course Objectives:

1. Introduction of Internet and basic components of Internet.
2. To explain the fundamentals of HTML and HTML 5.
3. To explain the basic properties of CSS and CSS3 in html file
4. To explain the basic properties of JavaScript in html file
5. To explain dynamic web page using JavaScript

Course Outcomes:

Upon successful completion of the course, the students will:

1. Apply the basics of Web Designing & Publishing.
2. Apply the implementation of HTML tags and be able to create HTML static web pages.
3. Able to use basic properties of JavaScript in html file
4. Able to create dynamic web page using JavaScript

UNIT I:

BASICS OF WEB DESIGN: Overview of Internet and WWW, Basic elements of the Internet, Internet services, Internet Browsers and Servers, Introduction to WWW, URL, webpage, web site, web

servers, web browser, Web Application, Client and Server-side scripting languages, types of websites, Web Design and Development, Internet Addressing: standard Internet Address, Domain Name Server (DNS).

UNIT II:

WEB PAGE DEVELOPMENT USING HTML5: HTML Fundamentals: HTML & its relevant history, Anatomy of an HTML Tag, Basic HTML Document Structure, working with Text, working with Lists, Tables and Frames,working with Hyperlinks, Images and Multimedia, Working with Forms and Controls. Advanced Elements in HTML5: Semantic Elements, New Input Type Elements, Multimedia Tags.

UNIT III:

INTRODUCTION TO CASCADING STYLE SHEETS: Introduction to Cascading Style Sheet (CSS), basic syntax and structure, CSS selectors, Ways of specifying style, CSS Properties, CSS Styling (Background, Text, Fonts, Lists, Tables, Links), CSS Box Model, CSS Navigation Bar; CSS3: - CSS Rounded Corners, Box & Text Shadow, Gradients, Background Images, Transitions, Transforms, and Animations, CSS Layout.

UNIT IV:

JAVA SCRIPT PROGRAMMING I: Introduction to Client-Side Scripting, Basics of Java Script, Java Script Statements, Comments, variables, Operators and Expressions, Conditions statement, Functions, Dialog boxes.

UNIT V:

JAVASCRIPT PROGRAMMING II**:** The Java Document Object Model (DOM): JavaScript Document Object Model hierarchy – Create, find and manipulate HTML Element using Objects and methods. Form validation, Applying Style using JavaScript. Creating New window, Accessing & manipulating, History of HTML Pages; Forms: Form object, built in objects, User defined objects, Cookies, Java Script Window; DHTML: Introduction to DHTML, DHTML CSS, DHTML Java Script, DHTML HTML DOM, DHTML Events.

Text Books:

1. Matthew MacDonald: HTML5 The Missing Manual, O'Reilly Media, August 2011.
2. Peter Gasston: The Book of CSS3, A Developer's Guide to the Future of Web Design, No Starch Press, April 2011.
3. Richard York: Beginning CSS Cascading Style sheets for Web Design, Wrox Press (Wiley Publishing), 2005.

Reference Books:

1. **Ivan** Bayros: Web Enabled Commercial Application Development using HTML, JavaScript, DHTML and PHP, fourth revised edition, BPB Publication.
2. David Mc Farland: CSS The Missing Manual, O'Reilly, 2006.
3. Julie C. Meloni: HTML, CSS and JavaScript All in One, Pearson.

Web References:

1. http://www.tutorialspoint.com/html5 [For notes on HTML5 tags]
2. http://www.w3schools.com/html/ [For HTML5, CSS and JavaScript notes and examples]
3. https://in.godaddy.com/help/dreamweaver-cs6-publish-your-website-7811[Publish your website using Dreamweaver]
4. http://fullbooksfreedownload.blogspot.in/2016/02/html-css-javascript-webpublishing-in.html [Book :- HTML, CSS & JavaScript Web Publishing in One Hour a Day, Sams Teach Yourself, 7th Edition PDF]

Course Code: CS-AEC-151 EL-151 **Course Title: ENGLISH-1**

Credits: 2 **LTP: 2– 0 – 0**

Course Code: CS-SEC-151 **Course Title: b) Lab on Python Programming**

Credits: 3 **LTP: 0– 0 – 6**

Course Objectives: The objective of the course is to make the students aware and proficient in Python Programming**.**

Course Outcome: The outcome of the course is that the students should be able to execute the broad class of problems as outlined in the syllabus of the lab on python programming.

1. Problems related to if ... .else structure of Python
2. Problems related to looping, break and continue
3. Problems to identify the usage of pass and range statements in python.
4. Problems related to usage of functions in Python, Global , local and Non local functions
5. Problems related to Recursion, Anonymous Function, Lambda function, Python Modules and Python Package.
6. Problems related to File handling in Python, exception handling, usage of User defined exception and Assertions
7. Problems related to Python Collections: List, Tuple, Sets and Dictionary
8. Problems related to String Manipulation, Basic Operations, Slicing, Python Regular expressions; Python iterators, Python Generators, Python Closure, Python Decorators, List Comprehension
9. Problems related to Python Object & Class , Problems on MATPLOTLIB
   Stack, queue implementation in Python, Implementing basic sorting algorithms like Merge Sort, Quick sort and stable sort
10. Problems related to Binary Search, Linked List and Binary Search trees

Course Code: CS-SEC-151 **Course Title: b) Laboratory on Numerical Methods**

Course Objectives:

The objectives of this course are to design the iterative algorithms and execute in the computer to solve various scientific numerical problems.

Course Outcomes:

Upon successful completion of the course, the students will:

1. Apply the structure of iterative algorithms that can be converted into computer program code.
2. Able to execute various numerical methods in the computer and will get a better understanding of these methods.
3. Improvement in the programming skills of the students

Problems related to scientific computation should be solved by using the high level programming language C (preferably on Unix/Linux/Solaris operating systems on a network). Following are some sample laboratory programming assignments but the assignments should not be limited to these only:

Write an algorithm and a C program to:

1. Find the roots of a given equation using:
    a. Bisection Method.
    b. Regula Falsi Method.
    c. Newton Raphson Method.
    d. Secant Method.
2. Find f(x) for a given set of experimental data using
    a. lagrange interpolation.
    b. Newton's forward interpolation.
    c. Newton's backward interpolation.
3. Fit a given set of data in a
    a. straight line
    b. parabola
    c. curve of the form $y=ax^2+b$
    d. curve of the form $y=ab^x$
    e. curve of the form $y=ae^{bx}$
4. Find the numerical solution of a system of linear equation using
    a. Gauss elimination
    b. Gauss Jacobi
    c. Matrix Inversion
    d. Gauss Seidel
5. Perform differentiation applying
    a. Taylor Series
    b. Euler's method
    c. Runge-Kutta method
    d. applying Laplace's equation
    e. Jacobi's Method
    f. Gauss seidel method
    g. Trapezoidal Rule
6. Perform numerical Integration applying Simpson's 1/3 Rule.

Course Code: CS-VAC-151 **Course Title: EVS**

Credits: 3 **LTP: - - -**

SEMESTER III

Course Code: CS-DSC-201 **Course Title: DATA STRUCTURE**

Credits: 4 **LTP: 3– 1 – 0**

Course Objectives:

This course will help students understand the fundamentals of data structures and algorithms also concepts of linear and nonlinear data structures, concepts of searching and sorting techniques and graph and file organizations.

Course Outcome:

After completing this course, student will be able to understand
1 The basic concepts of fundamental data structures and algorithms, array and link list. Also step by step approach in solving problems.
2.The basic concepts of stacks, queues,
3.Concepts various tree structures.
4.Concepts of searching and sorting techniques
5.Graph and graph search technique and file organizations

UNIT-I:
Introduction: Data Structures, Data Structures operation, Arrays, Multidimensional arrays, Representation of array in memory, address calculations, sparse arrays.
Lists: sequential and linking structures, circular lists, doubly linked lists, inverted lists, threaded lists, operations on all these structures and applications.

UNIT-II:
Stacks and Queues: Operations on Stack and Queues and their implementations, Applications of Stacks: Polish Notation, Applications of Queues, and Types of Queues: Priority Queue, Circular queue, Double Ended Queue, Implementation of stacks & queues using linked lists, Recursion.

UNIT-III:
Tree Structures: Introduction, binary trees, tree traversal algorithms, threaded trees, binary search trees, AVL search trees, B-trees.

UNIT-IV:
Sorting and Searching: Sequential Sort, Radix sort, Insertion Sort, Bubble Sort, Quick Sort, Merge Sort, Heap Sort Searching: Sequential Search and Binary Search.

UNIT-V:
Introduction to Graph and Graph Search Techniques, File Organization: serial, sequential, indexed sequential, direct inverted, multi-list, hashing and collision handling methods.

Text Books/ References:

1. Data structures using C -Tenenbaum, PHI, 1996

2. Fundamentals of Data Structures Horowitz Sahani, Computer Science Press, 1978
3. An introduction to data structures with applications Jean Paul Trembley and Paul Sorenson, McGraw Hill, International Student Edition, 1985
4. Data structures and Algorithms Aefred V. Aho, John E. Joperoft and J.E. Ullman
5. Data Structures, Seymour Lipschutz, Schaum's outlines, Tata McGraw Hill Education Private Ltd.

Course Code: CS-DSC-202	**Course Title: Computer Architecture**
Credits: 4	**LTP: 3– 1 – 0**

Course Objective:

This course aims to provide a comprehensive understanding of the fundamental hardware and software issues in computer organization, as well as an overview of the design principles governing digital computing systems. Additionally, the course will focus on the representation of data and computation at the machine level.

Course Outcome:

1. Understanding of the digital system, its organization and architecture.
2. Apply knowledge of digital electronics logic gates to combinational and sequential circuits.
3. Knowledge of the basics of computer hardware and how software interacts with computer hardware.
4. Apply concepts of assembly language in solving problems.
5. Illustrate the concept of processing I/O organization and examine different ways of communicating with I/O devices and standard I/O interfaces.

UNIT-I:
BASIC STRUCTURE OF COMPUTERS: Computer Types, Functional UNIT-s, Basic operational concepts, Bus structures, Software, Performance, multiprocessors and multi computers. REGISTER TRANSFER LANGUAGE AND MICROOPERATIONS: Register Transfer language. Register Transfer, Bus and memory transfer, Arithmetic Micro-operations, logic micro-operations, shift micro-operations, Arithmetic logic shift UNITs.

UNIT-II:
BASIC COMPUTER DESIGN- Instruction codes, Computer Registers, Computer instructions, Instruction cycle. Memory – Reference Instructions, Input – Output and Interrupt. MICRO PROGRAMMED CONTROL: Control memory, Address sequencing, micro program example, Design of control UNIT Hardwired control, Micro programmed control.

UNIT-III:
CENTRAL PROCESSING UNIT -Stack organization. Instruction formats. Addressing modes. DATA Transfer and manipulation. Program control. Reduced Instruction set computer. Parallel Processing, Pipelining.

UNIT-IV:
COMPUTER ARITHMETIC :Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. Decimal Arithmetic UNIT-s, Decimal Arithmetic operations.

UNIT-V:

INPUT-OUTPUT ORGANIZATION : Peripheral Devices, Input-Output Interface, Asynchronous data transfer Modes of Transfer, Priority Interrupt, Direct memory Access, Input –Output Processor (IOP), Serial communication; THE MEMORY SYSTEM : Memory Hierarchy, Main memory, Auxiliary memory, Associative memory, Cache memory, Virtual memory, Memory management hardware .

Text Books:

1. Computer System Architecture – M. Morris Mano, IIIrd Edition, PHI / Pearson, 2006.
2. Computer Organization – Car Hamacher, ZvonksVranesic, SafwatZaky, V Edition, McGraw Hill, 2002.

Reference:

1. Computer Organization and Architecture – William Stallings Seventh Edition, PHI/Pearson, 2006.
2. Computer Architecture and Organization – John P. Hayes, Mc Graw Hill International editions, 1998.
3. Introduction to computer architecture - Stones S. Galgotia Publication
4. Computer Engineering - Hardware Design - M. Morris Mano, PHI 5. Computer Architecture and parallel processing - Kai Hwang & Faye Briggs, McGraw hill, 1985

Course Code: CS-DSM-201, Course Title: Introduction to Probability and statistics

Credits: 4 **LTP: 3– 1 – 0**

Course Objectives:

The aim of the course is to enable the students with understanding of basic concepts of probability theory and various types of probability distributions. It aims to equip the students with standard concepts of statistical techniques and their utilization.

***Course Outcomes**:*

At the end of the course, the students will be able to:
I) Explore the basic ideas about measures of central tendency, dispersion and their applications in other statistical problems.
ii) Understand the basic concepts of Probability theory and its applications in various mathematical models.
iii) Explain the different types of discrete and continuous distributions and their utilizations.
iv) Apply the knowledge of Statistical techniques in various experimental and industrial requirements .

UNIT-I:

Measures of location, measures of dispersion, skewness, coefficient of skewness, Theory of probability, Axiomatic approach to probability, concept of events, sets, Additional multiplication theorem on probability, conditional probabilities, independent, pairwise independent and mutually independent events and applications, Bay’s theorems and applications, Laws of expectations, Moment Generating functions and variance-covariance matrix.

UNIT-II:
Random variables, Discrete and continuous, Probability mass function, probability of function, Joint distribution, P.D.F., conditional distribution and marginal distribution.

UNIT-III:
Theoretical discrete and continuous distributions, Binomial, Poisson, Normal, Beta, Exponential distribution, other discrete distributions (Derivations not necessary).

UNIT-IV:
Correlation, simple, partial and multiple correlations, regression, simple and complex regression, lines of regression, regressive curves and coefficients, Curve fitting by the least squares, Possible solution to system of linear equations by Lagrange’s principle squares.

UNIT-V:
Sampling, sampling of attributes, standard errors, sampling distribution, Testing of signal UNIT-izing X,T,F and Z-statics ,analysis of variance –one way and two way classes ,co-variance analysis

Text Books/References:

1. Fundamentals of Mathematical Statistics- S. C. Gupta, V. K Kapoor and Saxena ,1996, S Chand & Co. New Delhi
2. Mathematical statistics - Kapoor and Saxena, 1996, S.Chand & Co. New Delhi
3. Statistical methods - S. P. Gupta
4. Statistics - C. B. Gupta
5. Methods and Application -Sancheti and Kapoor
6. Fundamentals of Applied Statistics - S. C. Gupta and V. K. Kapoor

Course Code: CS-IDC-201 **Course Title: Cyber Security**
Credit: 3 **LTP: 3-0-0**

Course Objectives:
1.This course provides the foundation for understanding the key issues associated with protecting information assets.
2.The purpose of the course is to provide the student with an overview of the field of information security and assurance.

Course Outcomes:
1.Understand the broad set of technical, social & political aspects of Cyber Security.
2. Appreciate the vulnerabilities and threats posed by criminals, terrorists and nation states to national

infrastructure.
3.Understand the importance of ethical hacking tools.
4.Understanding the ethical hacking process.
5.Implementing ethical hacking tools in an organization.
6.Apply security principles to system design.
7.Apply methods for authentication, access control, intrusion detection and prevention and conduct research in Cyber Security

UNIT I:
INTRODUCTION: Introduction to Cyber Security - Importance and challenges in Cyber Security - Cyberspace - Cyber threats - Cyber warfare - CIA Triad - Cyber Terrorism - Cyber Security of Critical Infrastructure - Cyber security -Organizational Implications.

UNIT II:
HACKERS AND CYBER CRIMES: Types of Hackers - Hackers and Crackers - Cyber-Attacks and Vulnerabilities - Malware threats - Sniffing - Gaining Access - Escalating Privileges - Executing Applications - Hiding Files - Covering Tracks - Worms - Trojans - Viruses - Backdoors

UNIT III:
ETHICAL HACKING AND SOCIAL ENGINEERING : Ethical Hacking Concepts and Scopes - Threats and Attack Vectors - Information Assurance - Threat Modeling - Enterprise Information Security Architecture - Vulnerability Assessment and Penetration Testing - Types of Social Engineering - Insider Attack - Preventing Insider Threats - Social Engineering Targets and Defense Strategies.

UNIT IV:
CYBER FORENSICS AND AUDITING: Introduction to Cyber Forensics - Computer Equipment and associated storage media - Role of forensics Investigator - Forensics Investigation Process - Collecting Network based Evidence - Writing Computer Forensics Reports - Auditing - Plan an audit against a set of audit criteria - Information Security Management System Management. Introduction to ISO 27001:2013.

UNIT V:
CYBER ETHICS AND LAWS : Introduction to Cyber Laws - E-Commerce and E-Governance - Certifying Authority and Controller - Offences under IT Act- Computer Offences and its penalty under IT Act 2000 - Intellectual Property Rights in Cyberspace.

Text books:
1. Donaldson, S., Siegel, S., Williams, C.K., Aslam, A., “Enterprise Cyber security -How to Build a Successful Cyber Defense Program against Advanced Threats”, Apress, 1st Edition, 2015.
2. Nina Godbole, Sumit Belapure, “Cyber Security”, Willey, 2011.
3. Roger Grimes, “Hacking the Hacker” , Wiley, 1st Edition, 2017.
4. Cyber Law By Bare Act, Govt of India, It Act 2000

Course Code: CS-AEC-201 **Course Title: MIL-II**

Credit: 2 **LTP: 2-0-0**

Course Code: CS-SEC-201 **Course Title: Lab on Data Structure**

Credit: 3 **LTP: 0-0-6**

Course Objective:

The course is designed to develop skills to design and analyze simple linear and non linear data structures. It strengthens the ability of the students to identify and apply the suitable data structure for the given real world problem. It enables them to gain knowledge in practical applications of data structures.

Course Outcome:

At the end of this lab session, the student will

1.Be able to design and analyze the time and space efficiency of the data structure
2.Be capable to identity the appropriate data structure for given problem
3.Have practical knowledge on the applications of data structures

Write a C program to implement

1. Write a program to create a 1-D Array and display the elements of the array.
2. Write a Program to insert an element at the beginning/end/ anywhere in a 1-D array.
3. Write a Program Delete an element from beginning/end/ anywhere in a 1-D array.
4. Write a program to create a 2-D Array and display the elements of the array.
5. Write a program to display the non-zero elements of the sparse array.
6. Write a program using functions for implementation of circular Queue.
7. Write a program that uses functions and without function to perform the following operations on singly linked List (i) Creation (ii) Displays the element.
8. Write a program that uses functions to perform the following operations on singly linked List for Insertion of elements at the beginning/end/ anywhere.
9. Write a program that uses functions to perform the following operations on singly linked List for deletion of elements from the beginning/end/ anywhere.
10. Create a doubly linked list and display the elements.
11. Write a Program to Insert an element at the beginning/end/ anywhere in a doubly linked list.
12. Write a Program Delete an element from beginning/end/ anywhere in a doubly linked list.
13. Create a circular linked list and display the elements.

14. Write a Program to insert an element at the beginning/end/ anywhere in a circular linked list.
15. Write a Program Delete an element from beginning/end/ anywhere in a circular linked list.
16. Write a program to invert a singly linked list.
17. Write a program that implements stack (its operations) using (i)Arrays (ii)Linked list.
18. Write a program that implements Queue (its operations) using (i)Arrays (ii)Linked list.
19. Write a program that to perform the following searching operations for a Key value in a given list of integers: a) Linear search b) Binary search
20. Write a program that implements the following sorting 1. Bubble sort 2. Selection sort .
21. Write a program that implements the Quick sort.
22. Write a program that implements the following 1. Insertion sort 2. Quick Sort
23. Write a program that implements the following Heap sort.
24. Write a program that implements the following Merge sort .
25. Write a program that implements the following Radix sort.
26. Write a program to perform the following operations: a) Insert an element into a binary search tree. b) Delete an element from a binary search tree. c) Search for a key element in a binary search tree.
27. Write a program using functions for binary tree traversals Pre-order, In-order and Post-order using both Recursive and Non-recursive approach.
28. Write a Program in C for implementation of Insertion and Deletion in BST Using Linked List.
29. Write a program to perform the following operations: a) Insert an element into an AVL tree. b) Delete an element from an AVL tree. c) Search for a key element in an AVL tree.
30. Write a program to perform the following operations: a) Insert an element into an AVL tree. b) Delete an element from an AVL tree. c) Search for a key element in an AVL tree
31. Write a program in C to implement depth first search.
32. Write a program in C to implement Breadth first search.

SEMESTER IV

Course Code: CS-DSC-251 **Course Title: Database Management System**

Credit: 4 **LTP: 3– 1 – 0**

Course Objectives:

1. To present necessary concepts for database designing.
2. To introduce how to evaluate a set of query using SQL and algebra.
3. To build Concepts of RDBMS, and learn Object oriented modeling.
4. To introduce transaction management and concurrency.
5. To introduce storage structure and file management.
6. To introduce query optimization and query processing.
7. To introduce data mining and data warehousing.

Course Outcomes:

1. Understand concepts of database system architecture.
2. Design conceptual, logical database model and physical model.
3. Understand relational models and perform SQL operations.
4. Understand the importance of normal forms and learn query optimization.
5. Learn the importance of transaction processing and concurrency control.
6. Create, test and validate a database.

UNIT-I:
Data modeling for a database: records and files, abstraction and data integration, database management systems: the three levels architecture of a DBMS, components of a DBMS.

UNIT-II:
Data models: Hierarchical, Network model, Relational; ER Diagrams, Extended ER Diagrams, Database Schema, Keys, Relational Database: manipulations; relational algebra.

UNIT-III:
Relational calculus, SQL Queries, Relational database design

UNIT-IV:
Normalization and Functional dependencies, findings keys, decomposition computing closures of a FD's Query processing: general strategies for query processing and query optimization, query processor.

UNIT-V:
Transactions and Transaction Processing, ACID Properties, Introduction to Concurrency and Serialization, Concepts of Security and Recovery.

Text books/References:

1. Fundamentals of Database System: R. Elmasri & S. Navathe (Benjamin Cummings).

2. Database Management System-Henry F. Korth & Abraham Silberschats, McGraw hills, 1991.
3. An introduction to database management system vol I &II-Date C.J., Addison Wesley, 1981, 1983
4. Principles of database system -Ullman J.D., computer science, 1982.

Course Code: CS-DSC-252 **Course Title: Microprocessor**

Credit: 4 **LTP: 3– 1 – 0**

Course Objectives :

1. To learn about various terminologies related to microprocessor.
2. To understand the various functional units of computer.
3. To understand the architecture of 8086 Microprocessor
4. To learn about the various units of ALU.
5. To understand about various types of addressing modes.
6. To learn the various instruction set of 8086 processor, so that they will be able to do assembly language programming.
7. To perform different operations on data using assembly language programming.
8. To acquire the knowledge of Interfacing of I/O and Memory peripherals with microprocessor.
9. To Design few sample applications using assembly language programming.

Course Outcomes:

1. Students will be familiar with functional units of CPU.
2. They will be aware about architecture of 8086 Microprocessor.
3. They will have knowledge of flowchart and assembly language programming.
4. They will have understanding about various types of instruction formats and addressing modes.
5. They will be able to do different operations on data using assembly language programming.
6. They will be familiar with the concept of Interfacing of I/O and Memory peripherals with microprocessor.
7. They will be able to make sample applications using 8086 assembly language program.

UNIT-I:

Architecture of 8086 Microprocessor. Special functions of General purpose registers. 8086 flag register and function of 8086 Flags. Addressing modes of 8086. Instruction set of 8086. Pin diagram of 8086-Minimum mode and maximum mode of operation. Timing diagram. Memory interfacing to 8086 (Static RAM & EPROM).Need for DMA. DMA data transfer Method. Interfacing with 8237/8257.

UNIT-II:

Assembler directives, simple programs, procedures, and macros. Assembly language programs involving logical, Branch & Call instructions, sorting, evaluation of arithmetic expressions, string manipulation.

UNIT-III:

8255 PPI – various modes of operation and interfacing to 8086.Interfacing Keyboard, Displays, 8279 Stepper Motor and actuators. D/A and A/D converter interfacing.

UNIT-IV:

Interrupt structure of 8086. Vector interrupt table. Interrupt service routines. Introduction to DOS and BIOS interrupts. 8259 PIC Architecture and interfacing cascading of interrupt controller and its importance.

UNIT-V:

Serial data transfer schemes. Asynchronous and Synchronous data transfer schemes. 8251 USART architecture and interfacing.TTL to RS232C and RS232C to TTL conversion.Sample program of serial data transfer. Introduction to High-speed serial communications standards, USB.

Text books:

1. Advanced microprocessor and Peripherals - A.K.Ray and K.M.Bhurchandi, TMH, 2000.
2. Microcontrollers – Deshmukh, Tata McGraw Hill Edition.
3. Microprocessors Architecture, Programming and Applications-Ramesh S. Gaonkar, Wiley eastern, 1994 or latest edition.

References :

1. Micro Processors & Interfacing – Douglas U. Hall, 2007.
2. The 8088 and 8086 Microprocessors – PHI, 4th Edition, 2003.
3. Microcomputer System 8086/8088 Family Architecture, Programming and Design - By Liu and GA Gibson, PHI, 2nd Ed.
4. Introduction to microprocessors -Aditya P. Mathur, TMH, 1995

Course Code: CS-DSC-253	**Course Title: Discrete Mathematics**
Credit: 4	**LTP: 3– 1 – 0**

Course Objectives:

Prepare students to develop Mathematical foundations to understand and create mathematical arguments required in learning many Mathematics and Computer Science courses. To motivate students how to solve practical problems using discrete Mathematics. Also in this course, the basic concepts of Graph theory are introduced.

Course Outcomes:

At the end of the Course, the students will be able to:
i) Construct mathematical arguments using logical connectives and quantifiers.
ii) Understand how lattices and Boolean algebra are used as tools and mathematical models in the study of Networks.
iii) Learn how to work with some of the discrete structures which include sets, relations, functions, graphs and recurrence relation.

UNIT-I:
Mathematical Logic: Propositional logic, truth table, propositional equivalence, Argument, Predicates and Quantifiers, Nested quantifiers, Rules of inference, Inference theory of the predicate Calculus.

UNIT-II:
Abstract Algebra: Group, Subgroup, Semigroup, Abelian group, Cyclic group, Lagrange theorem, Normal subgroup, Automorphisms, Homomorphisms, Permutation groups.
Rings: Definition and examples of rings, Simple properties of ring, Integral domain, Skew fields and Field.

UNIT-III:
Lattices and Boolean Algebra with applications: Lattices and partially ordered sets, Bounded lattice, Distributive lattice, Complemented lattice, Boolean Algebra, Boolean functions, Logic gates, Representation and Minimization of Boolean functions and Finite state Machines.

UNIT-IV:
Graph Theory: Basic concepts of graph theory, complete graphs, bipartite graphs, subgraph, Adjacency matrix representation of graph, Incidence matrix of a graph, connected graph, Euler paths and circuits, Hamilton paths and circuits, Dijkstra’s Algorithm for shortest path, planar and nonplanar graphs.

UNIT-V:
Trees: Introduction to trees, Rooted tree, Binary tree, Decision tree, Tree Traversal (preorder, inorder, post order traversal), Spanning tree, Depth first search, Breadth first search, Minimum spanning trees, Prim’s Algorithm.

Text Books/ References:

1. Discrete Mathematics & Its Applications: Kenneth H Rosen, Tata McGraw Hill, Sixth Edition.
2. Topics in Algebra: I.N. Herstein, Wiley Student Edition, Second Edition.
3. Discrete Mathematics: Richard Johnsonbaugh, Pearson Education, Fifth Edition.

Course Code: CS-DSM-251	**Course Title: Lab on a) DBMS**
Credit: 3	**LTP: 0-0-6**

Course Objectives:

1. To practice the gaining knowledge of DBMS paper by developing a database.
2. To introduce MySQL/ SQL/ SQL Server/ PLSQL/ Oracle etc.
3. To introduce the designing, developing and querying a database.

Course Outcomes:

1. To design and implement a database schema for a given problem.
2. Students should solve assignments by using the standard principles and practices of relational database design and then develop the appropriate schema for machine implementation on MySQL/SQL/SQL Server/PLSQL/Oracle etc. in Windows/Unix/Linux/Solaris operating systems environment on a network.
3. To develop applications for real-life problems applying the normalization techniques
4. To formulate and evaluate queries using SQL DML/DDL/DCL commands
5. To Create, test and validate a database..

Following are some samples for laboratory programming assignments but the assignments should not be limited to these only. These programming assignments must be preceded by corresponding database design assignments.

1. Creating Database: Creating a Database, Creating a Table, Specifying Relational algebraic constructs, Specifying Constraints, Creating Indexes.

2. Table and Record Handling: INSERT statement, Using SELECT and INSERT together, DELETE, UPDATE, TRUNCATE statements, DROP, ALTER statements.

3. Retrieving Data from a Database: The SELECT statement, Using the WHERE clause, Using Logical Operators in the WHERE clause, Using IN, BETWEEN, LIKE , ORDER BY, GROUP BY and HAVING clause, Using Aggregate Functions, Combining Tables Using JOINS, Sub queries.

4. Database Management: Creating Views, Creating Column Aliases, Creating Database Users, Using GRANT and REVOKE, Cursors in Oracle PL / SQL, Writing Oracle PL / SQL Stored Procedures.

Course Code: CS-DSM-251 **Course Title: Lab on b) Microprocessor**

Credit: 3 **LTP: 0-0-6**

Course objectives:

1.To provide an in-depth understanding of the operations of microprocessors, assembly language programming and interfacing.
2.To provide an in-depth understanding of the operations of basic microcontrollers.

Course Outcomes:

1. Understand the architecture and working principles of microprocessors.
2. Write a program in assembly level language.
3. Acquire skills in memory and peripheral interfacings.
4. Gain knowledge of microcontrollers and applications.
5. Solve real-world problems.

Following are some sample laboratory programming assignments but the assignments should not be limited to these only.

1. Write an assembly language Program to perform different modes of operation (HEX KEYPAD/ Serial modes).
2. Write an assembly language Program to examine a series of memory byte locations from 0:1234.
3. Write an assembly language Program to examine and modify memory word location from 500:340.
4. Write an assembly language Program to examine registers.
5. Write an assembly language Program to modify registers DX to 55AA
6. Executing programs stored at memory location 1000H
7. Write an assembly language Program to convert a hexadecimal number to decimal number.
8. Write an assembly language Program to convert decimal to hexadecimal numbers.
9. Write an assembly language Program for hexadecimal addition/subtraction
10. Write an assembly language Program for hexadecimal multiplication/division

Course Code: CS-DSM-252 Course Title: Data Communication and Computer Networks

Credit: 3 LTP: 3-0-0

Course Objectives:

1.To give clear idea of signals, transmission media, errors in data communications
2.Brief the students regarding protocols and standards.
3.To Focus on different coding schemes.
4.To introduce flow of data, categories of network, different topologies.
5.To Focus on information sharing and networks, their correction, networks classes and devices,etc

Course Outcomes:

1.On successful completion of the course, the student will have the basic knowledge of computer networks.
2.Knowledge of data sharing, transmission media and their protocols.
3.Knowledge of various types and means of transmission medium.
4.Knowledge of error detection and correction Techniques.

UNIT-I:

Introduction to data communication and networking: Why study data communication?, Data Communication, Networks, Protocols and Standards, Standards Organizations. Line Configuration, Topology, Transmission Modes, Categories of Networks Internet works.

UNIT-II:

Study of OSI and TCP/IP protocol suit: The Model, Functions of the layers, TCP/IP Protocol Suites. **Study of Signals:** Analog and Digital, Periodic and Aperiodic Signals, Analog Signals, Time and Frequency Domains, Composite Signals, Digital Signals.

UNIT-III:

Study of Digital and analog transmission: Digital to Digital Conversion, Analog to Digital Conversion.

Introduction to networks and devices: Network classes, Repeaters, Hub, Bridges, Switches, Routers, Gateways Brouters Routing Algorithms, Distance Vector Routing, Link State Routing.

UNIT-IV:

Study of Multiplexing: Many to one/one to Many, Frequency division Multiplexing, Wage division Multiplexing, Time division Multiplexing, Multiplexing applications. **Types of transmission media**: Guided Media, Unguided Media, Transmission Impairments, Performance Wavelength, Shannon Capacity, Media Comparison, PSTN, Switching

UNIT-V:

Error Detection and Correction: Types of Errors, Detection, Parity Check, Vertical Redundancy Check, Longitudinal Redundancy Check, Cyclic Redundancy Check, Checksum, Error Correction.

Text Books:
1. Data communication & Networking by Bahrouz Forouzan.
2. Computer Networks by Andrew S. Tanenbaum

Reference Books:
1. Data and Computer Communications by William Stallings

Course Code: CS-AEC-251	**Course Title: English-II**
Credit: 2	**LTP: 2-0-0**

SEMESTER V

Course Code: CS-DSC-301	**Course Title: Operating System**

Credit: 4 **LTP: 3– 1 – 0**

Course Objectives:

The main objective of this course is to discuss the basic architecture of the operating systems. The course covers various important modules of operating systems and their functionalities.

Course Outcomes:

At the end of semester, Student will able to:
1. Understand the underlying design of operating system architecture.
2. learn basic concepts of process management, memory management and the effect of different algorithms to handle multiple processes and to allot memory amongst them.
3.understand the basic principle of secondary storage management and the organization of files in computer storage devices.
4. learn the basic challenges and mechanisms to provide a co-operative environment to multiple processes.

UNIT-I:

Operating System Architecture: Operating System as an extended machine and resource manager, Operating system classification, Operating system modes and System calls, Processor management functions: process model, process states and transitions, multiprogramming, multitasking, multithreading, CPU scheduling algorithms

UNIT-II:

Memory management functions: memory management of a single user operating system, memory management for multi user operating systems, Logical versus Physical address space, Swapping, Paging, Segmentation, Segmentation with paging, virtual memory.

UNIT-III:

File Management functions: File naming structure , access mechanism , attributes and operations , Directory structure , file protection and security , File space Allocation methods , File sharing , file locking , symbolic links , distributed file system.

UNIT-IV:

Concurrent Programming: Sequential and Concurrent Processes, The Critical Section Problem,Semaphores , Classical Problems of Synchronization , Critical Regions , deadlock handling , inter process communication.

UNIT-V:

Device Management functions: I/O device and controllers, device drivers, interrupt servicing, direct memory access ,clocks and timers , disk scheduling , Protection: Goal, principles and domains of protection, Access matrix, access control, Revocation of access right;

Text books/References:

1. Operating System Concepts,Silberschatz , Galvin , John wiley and sons
2. Operating Systems,Stallings , Pearson
3. Operating Systems, H.M. Deittel, Pearson
4. Systems Programming,D . M. Dhamdhere , McGraw Hill

Course Code: CS-DSC-302 **Course Title: System Software**

Credit: 4 **LTP: 3– 1 – 0**

Course Objectives:

The objective of this course is to design the basic architecture of computer system programs. This course covers the different modules to convert a high level language to machine level language.

Course Outcomes:

At the end of this course, student will able to

1. Understand the basic concepts of system software and the importance of system program.
2. Learn to write assembly level program code and the conversion into the machine level code.
3. Learn the dynamic allocation of memory to program code and linking and loading these program code.
4. Get the basic knowledge on the architecture and functionalities of a compiler.

UNIT-I:

Introduction, System Software and Machine Architecture. Simplified instructional Computers (SIC)-SIC machine architecture, Data and instruction formats - addressing modes - instruction sets - I/O and programming, SIC /XE Machine Architecture, SIC programming Examples, machine structure with special reference to IBM 360 and 370 systems.

UNIT- II:

Assembler: Definition, general design procedures ,A simple SIC assembler, Assembler Algorithm & Data Structures , Machine dependent assembler features - Instruction formats and addressing modes , Program relocation, Machine independent assembler features - Literals – Symbol-defining statements , Expressions, Program Blocks, Control Sections and Program Linking. One pass assemblers and Multi pass assembler, Implementation example - MASM assembler.

UNIT-III:

Basic macro processor functions - Macro Definition and Expansion, Macro Processor Algorithm and data structures , Machine-independent macro processor features - Concatenation of Macro Parameters, Generation of Unique Labels, Conditional Macro Expansion, Keyword Macro Parameters, Macro Processor design Options- Recursive Macro Expansion. General purpose Macro Processors, Macro Processing within Language Translators, Macro-Implementation example - MASM Macro Processor – ANSIC Macro language.

UNIT-IV:
Basic loader functions, Design of an Absolute Loader , A Simple Bootstrap Loader, Machine dependent loader features – Relocation, Program Linking, Algorithm and Data Structures for Linking Loader, Machine-independent loader features - Automatic Library Search , Loader Options, Loader design options , Linkage Editors, Dynamic Linking, Bootstrap Loaders - Implementation example - MS-DOS linker.Text Editors - Overview of Editing Process, User Interface, Editor Structure, Interactive Debugging Systems-Debugging Functions and Capabilities, Relationship With Other Parts Of The System

UNIT-V:
Basic Compiler Functions: Grammars, Lexical analysis, Syntactic analysis, Code Generation. Machine Dependent Compiler Features: Intermediate form of the program, Machine Dependent Code Generation Machine Independent Compiler Features: Structured variable, Machine Independent Code Generation, Storage Allocation, Block -Structured Languages. Lex and Yacc- The Simplest Lex Program, Recognizing Words with LEX, Symbol Tables, Grammars, Parser –Lexer Communication, The Parts of Speech Lexer, A YACC Parser, The Rules Section, Running LEX and YACC. LEX vs. Hand – Written Lexers.

Text books/references:

1. System Software by Leland. L. Beck, Pearson Education
2. Lex and Yacc by John.R.Levine, Tony Mason and Doug Brown, O'Reilly, SPD
3. Systems Programming – John J. Donovan, Tata Mc Graw Hill (Latest edition)
4. Introduction to Systems Software – D. M. Dhamdhere, Tata Mc Graw Hill (Latest edition)

Course Code: CS-DSC-303	**Course Title: Computer Graphics**
Credit: 4	**LTP: 3– 1 – 0**

Course Objective:

1. To explain the basics of computer graphics and its applications.
2. To explain the use of co-ordinate system in graphics.
3. To explain the various algorithms for drawing structures and shapes.
4. To explain the hardware and software related to computer graphics.
5. To explain the design of graphics & animations.

Course Outcome:

Upon completion of this course, the students will be able to:
1.Design graphics and animations for any field of study.
2.Write graphics programs for any front-end design.
3.Perform basic projects in graphics.

4.Conceive the idea of 3D graphics & views.
5.Extend the application of graphics for other image processing areas.

UNIT-I:
Development of computer graphics: Basic graphics systems and standards, Coordinate systems, Raster scan and Random scan graphics, Display processors and display controllers, Color display techniques,Color spaces, Overview of color models (RGB, CMY) Frame buffer and bit operation, Concept in raster graphics.

UNIT-II:
Points, lines and curves, Scan Conversion, Line drawing algorithms (DDA, Bresenham's), Circle drawing algorithms (DDA, Bresenham's) and Ellipse generation, Polygon-filling, Conic-section generation, Antialiasing.

UNIT-III:
Two-dimensional viewing: Basic transformations, Windowing and clipping, Segments, Interactive picture construction techniques, Interactive input/output devices.

UNIT-IV:
Three dimensional Concepts: 3-D representations and transformations, Spline curve and surfaces, Fractals, 3-D viewing, Algorithm for 3-D volumes, Hidden lines and Surface rendering.

UNIT- V:
Multimedia and Animations, Introduction to Graphics packages and applications, Morphing, Overview of Graphics programming, Keyboard and Mouse Handling Programs.

Text books:

1. Hearn D., Baker, Computer Graphics with OpenGL, 3rd Edition, Pearson, 2009
2. Malay K. Pakhira, Computer Graphics, Multimedia and Animation, 2nd Edition, PHI, 2010

Reference books:

1. D. F. Rogers and J. A. Adams, Mathematical Elements of Computer Graphics, 2nd Edition, TMH, 1990
2. P. Shirley, M. Ashikhmin, S. Marschner, Fundamentals of Computer Graphics, 3rd Edition, CRC Press, 2009.

E – BOOKS

1. https://math.hws.edu/eck/cs424/downloads/graphicsbook-linked.pdf
2. https://ebooks.lpude.in/computer_application/mca/term_3/DCAP504_COMPUTER_GRAPHICS.pdf

MOOCs

1. https://onlinecourses.nptel.ac.in/noc22_cs111
2. https://www.coursera.org/learn/interactive-computer-graphics

Course Code: CS-DSM-301	**Course Title: a)Lab on Operating System**
Credit: 3	**LTP: 0-0-6**

Course Objectives:

The objective of laboratory paper on operating systems is to learn handling of computer processes, memory, secondary storage management and synchronization practically. Students will execute different algorithms to perform process scheduling, memory management and synchronization. Students will also learn various commands to access the linux operating system.

Course Outcomes:

1. Student will be familiarized with linux environment and will learn to execute shell programming
2. Students will be able to understand the procedure to create a process, modify a process and kill a process.
3. Students will be able to compare the performance of different process scheduling algorithms, page replacement algorithms by implementing and executing on the computer.
4. Students will be able to configure Server and security mechanisms.

Experiments:
Problems related to Operating Systems and Architecture (with Unix/Linux/Solaris) should be solved by using Programming languages C/C++/ JAVA (preferably on Unix/Linux/Solaris operating systems environment on a network). Further shell programming in UNIX/LINUX should be performed. Following are some areas of Operating Systems and Architecture (with Unix/Linux/Solaris) for laboratory programming assignments/experiments but the assignments should not be limited to these only.

1. Shell Programming
2. Process Management (fork(), exec(), wait(), exit() etc…)
3. Process Scheduling (FCFS, SJF, Priority, RR)
4. Memory Management(FIFO, LRU, OPT)
5. Threading
6. Server configuration (FTP, SMTP, DNS)
7. NFS Configuration
8. Firewall Configuration using iptables/ipchains(Linux only)

Course Code: CS-DSM-301	**Course Title: b) Lab on Computer Graphics**
Credit: 3	**LTP: 0-0-6**

Course Objective:

1.To explain the basics of computer graphics and its applications.
2.To explain the components of the graphical part in a computer system.
3.To explain the methods to plot and trace various shapes like line, circle, ellipse etc.
4.To explain the 2D and 3D transformations and viewing.
5.To explain the advanced graphics and animation including shading, rendering etc.

Course Outcome:

Upon completion of this course, the students will be able to:

1. Write programs to display various shapes like line, rectangles, circles, ellipse etc.
2. Perform the transformations of objects using translation, scaling and rotation cases.
3. Design basic animations.
4. Create and execute graphical models.
5. Extend the use of graphical tools for various applications like photoshop, blender, VFX.

Languages to be used: C/ C++/Java/ Any other suitable programming languages

List of Programs:

1. Write a program to draw a line using the DDA algorithm.

2. Write a program to draw a line using Bresenham's algorithm.

3. Write a program to draw a circle using the DDA algorithm.

4. Write a program to draw a circle using the midpoint circle algorithm.

5. Write a program to draw an ellipse line using a midpoint ellipse algorithm.

6. Write a program to draw a rectangle (100 pixels x 50 pixels) using lines.

7. Write a program to draw a square (100 pixels x 50 pixels) using lines.

8. Write a program to draw a toy house using lines.

9. Write a program to fill a rectangle using a boundary fill algorithm.

10. Write a program to fill a rectangle using a flood fill algorithm.

11. Write a program to simulate the traffic light operation.

12. Write a program to draw a 2D square and show its movement using 2D transformations.

13. Write a program to draw a 3D cube and show its movement using 3D transformations.

14. Write a program to design and implement a small video game like tic tac toe.

15. Write a program to design and implement an animation model using a blender tool.

Text Books:

1. Hearn D., Baker, Computer Graphics with OpenGL, 3rd Edition, Pearson, 2009

2. Malay K. Pakhira, Computer Graphics, Multimedia and Animation, 2nd Edition, PHI, 2010

Reference Books:

1. D. F. Rogers and J. A. Adams, Mathematical Elements of Computer Graphics, 2nd Edition, TMH, 1990
2. P. Shirley, M. Ashikhmin, S. Marschner, Fundamentals of Computer Graphics, 3rd Edition, CRC Press, 2009

E-books:

1. https://math.hws.edu/eck/cs424/downloads/graphicsbook-linked.pdf

2.https://ebooks.lpude.in/computer_application/msc_it/term_3/DCAP504_COMPUTER_GRAPHICS.pdf

MOOCs:

1. https://onlinecourses.nptel.ac.in/noc20_cs90/preview
2. https://www.coursera.org/learn/interactive-computer-graphics

Course Code: CS-DSM-302 **Course Title: a) Simulation and Modeling**

Credit: 3 **LTP: 3-0-0**

Course Objective and Outcomes :

The learners will have

1. A knowledge and understanding of the basic structure of a model, components of a model, functioning of a model etc.
2. A knowledge and understanding of representation of world or real time events in a computer both quantitatively and qualitatively.
3. A knowledge and understanding of starting a simulation or a prototype of a particular subject area.
4. A knowledge and understanding of discrete event simulations and continuous event simulations, their properties, design and implementations.

UNIT-I:

Introduction: When simulation is the appropriate tool and when it is not appropriate; Advantages and disadvantages of Simulation; Areas of application, Systems and system environment, Components of a system, Discrete and continuous systems, Model of a system; Types of Models, Discrete-Event System Simulation; Steps in a Simulation Study. The basics of Spreadsheet simulation, Simulation example: Simulation of queuing systems in a spreadsheet.

UNIT-II:

General Principles, Simulation Software: Concepts in Discrete-Event Simulation: The Event-Scheduling /

Time-Advance Algorithm, World Views, Manual simulation Using Event Scheduling, List processing. Simulation in Java, Simulation in GPSS

Statistical Models in Simulation: Review of terminology and concepts, Useful statistical models, Discrete distributions; Continuous distributions, Poisson process, Empirical distributions.

UNIT-III:

Queuing Models: Characteristics of queuing systems; Queuing notation; Long-run measures of performance of queuing systems; Steady-state behavior of M/G/1 queue; Networks of queues; Rough-cut modeling: An illustration. Queuing models – single and multi server queues, steady state behaviour of queues, Inventory system simulation.

UNIT-IV:

Random-Number Generation, Random-Variate Generation: Properties of random numbers; Generation of pseudo-random numbers; Techniques for generating random numbers; Tests for Random Numbers Random-Variate Generation: Inverse transform technique; Acceptance-Rejection technique; Special properties.

Input Modeling: Data Collection; Identifying the distribution with data; Parameter estimation; Goodness of Fit Tests; Fitting a non-stationary Poisson process; Selecting input models without data; Multivariate and Time-Series Input models

UNIT-V:

Estimation of Absolute Performance: Types of simulations with respect to output analysis; Stochastic nature of output data; Absolute measures of performance and their estimation; Output analysis for terminating simulations; Output analysis for steady-state simulations.

Verification, Calibration, and Validation; Optimization: Model building, verification and validation; Verification of simulation models; Calibration and validation of models, Optimization via Simulation

Text Books:

1. Jerry Banks, John S. Carson II, Barry L. Nelson, David M. Nicol: Discrete-Event System Simulation, 5th Edition, Pearson Education, 2010. (Listed topics only from Chapters1 to 12)
2. NarsingDeo “System Simulation with Digital Computer” PHI pub.

Reference Books:

1. Lawrence M. Leemis, Stephen K. Park: Discrete – Event Simulation: A First Course, Pearson Education, 2006.
2. Averill M. Law: Simulation Modeling and Analysis, 4th Edition, Tata McGraw-Hill, 2007.

Course Code: CS-SEC-301 **Course Title:Internship/community engagement/field study**
Credit: 2 **LTP: 0-0-4**

SEMESTER VI

Course Code: CS-DSC-351 **Course Title: Object Oriented Programming with C++**

Credit: 4 **LTP: 3-1-0**

Course Objectives:

1. Introduction of object-oriented programming and basic components of C++.
2. To explain the polymorphism and implementation of polymorphism in C++.
3. To explain the inheritance and implementation of inheritance in C++.
4. To explain the exception handling and type casting operators, is implementation in C++.
5. To explain the managing, I/O formats and implementation in C++.

Course Outcomes:

Upon successful completion of the course, the students will:

1. Understand the basic structure of C++ program and understand the concept of object-oriented programming characteristics.
2. Understand the concept of Constructor and Destructor and be able to create class and objects.
3. Understand the need of Polymorphism and implement static polymorphism.
4. Understand the purpose of Inheritance, identify relations between classes, implement inheritance and perform dynamic binding.
5. Able to handle the runtime errors or exceptions in C++ and use type cast operators. Able to read input from device, file, command line arguments and apply formats to input and output.

UNIT - I:

INTRODUCTION TO OBJECT ORIENTED PROGRAMMING : Structured programming vs. object-oriented programming, Introduction to C++: structure of a C++ program Tokens, inline function, pass by reference, default arguments, OOP characteristics. Define class, objects, visibility modes, static members, friend function, Constructors and Destructors, Default Constructor, Copy constructors, Parameterized Constructor, this keyword.

UNIT - II:

POLYMORPHISM : Introduction of Static Binding, Function Overloading and ambiguity raised in function overloading; Operator overloading through member function and friend functions, type conversions from primitive to class, class to primitive and class to class.

UNIT - III:

INHERITANCE :Objective of Inheritance, types of inheritance, impact of access modes during inheritance, Virtual base classes, Dynamic Binding, Virtual functions, Pure virtual functions, Abstract class, early vs. late binding.

UNIT - IV:

EXCEPTION HANDLING AND TYPE CAST OPERATORS: Introduction to Exception handling, catching class types, using multiple catch statements, exception handling options, creating custom exceptions, introduction to cast operators, const cast, static_cast, reinterpret_cast, dynamic cast.

UNIT - V:

MANAGING I/O FORMATS AND OPERATIONS: Understand and apply concepts from various header files like iostream. Iomanip ,fstream, etc. opening and closing files, reading and writing text files and binary files, performing random access on files, and handling command line arguments.

Core Books:

1. Herbert Schildt: C++: A Beginner's Guide, 2nd Edition Paperback, McGraw-Hill,2003.
2. E Balagurusamy: Object Oriented Programming with C++ Paperback, McGraw Hill India,2017.

Reference Books:

1. Yashavant Kanetkar: Let Us C++, Paperback, BPB Publications,2020.
2. Robert Lafore: Object Oriented Programming in C++, 4th Edition, Sams Publications,2002.
3. Bruce Eckel:Thinking in C++, Volume 1, 2nd Edition, Pearson Education,2006.

Web References:

1. https://www.geeksforgeeks.org/c-plus-plus/ [For Entire Syllabus]
2. https://google.github.io/styleguide/cppguide.html [For coding standards and concepts]
3. http://www.tutorialspoint.com/cplusplus/cpp_basic_input_output.htm [For Practicals]

Course Code: CS-DSC-352 **Course Title: Programming in JAVA**

Credit: 4 **LTP: 3-1-0**

Course Objectives:

1. To explain how to create a class and how to implement the 1-D array, 2-D array, matrix addition, subtraction, multiplication, and division operations, demonstrate polymorphism, method overloading, default constructor, parameterized constructor, single level inheritance, multilevel inheritance using super and without super keyword using java programming.
2. To explain the implementation of exception handling using try, catch, throw, throws, and finally methods, create single thread, multi- thread using thread class and using runnable interface and also implement the join, is alive method along with priority of a thread using java programming.
3. To explain the implementation of event handling on keyboard, mouse, button, check box, radio button, list, choice, text field using frame class using java programming.

Course Outcomes:

1. Learn the implementation of 1-D array, 2-D array, matrix addition, subtraction, multiplication, and division operations, demonstrate polymorphism, method overloading, default constructor ,parameterized constructor, single level inheritance, multilevel inheritance using super and without super keyword using java programming.
2. Learn the implementation of exception handling using try, catch, throw, throws, and finally methods, create

single thread, multi- thread using thread class and using runnable interface and also implement the join, is alive method along with priority of a thread using java programming.
3. Learn the implementation of event handling on keyboard, mouse, button, checkbox, radio button, list, choice, text field using frame class using java programming.

UNIT-I:
Introduction to Java, Basic Features, Java Virtual Machine Concepts, A Simple JavaProgram, Primitive Data Type and Variables, Java Keywords, Integer and Floating-PointData Type, Character and Boolean Types, Declaring and Initialization Variables, Typecasting, Java Operators, Expressions, control statements, Arrays.

UNIT-II:
Class Fundamentals, Creating objects, Assigning object reference variables, Introducing Methods, Method overloading, Static methods, Constructors, overloading constructors, This Keyword, Using Objects as Parameters, Argument passing, Returning objects, Method Overriding, Garbage Collection, The Finalize( ) Method, Inheritance Basics, Access Control ,Multilevel Inheritance, Abstract Classes ,Polymorphism, Final Keyword, Defining Package, Interfaces, Implementing Interfaces, Interface and Abstract Classes.

UNIT-III:
Exception Handling-try, catch, throw, throws, finally, Multithreaded Programming- Extends Thread class, Runable interface, join and is alive method, I/O in Java, Text Streams, Buffered Stream, Print Stream, Random Access File, The String Class, String, Buffer Class and Methods.

UNIT-IV:
Building User Interface with AWT, Swing-based GUI, Layouts and Layout Manager, Container, set bounds method, frame class.

UNIT-V:
Event handling – Text field, Button, Choice List, check box, Radio button, Text area, Java Database Connectivity.

Text books/references:
1. Timothy Budd, An Introduction to Object Oriented Programming, Addison WesleyPublishing company (for UNIT--I).
2. Herbert Schildt, The complete Reference, Tata McGraw Hill Publishing company(10th Edition)
3. Patrick Naughton and Herbert Schildt, JAVA: the complete Reference, TataMcGraw-Hill Publishing company

Course Code: CS-DSC-353	**Course Title: Wireless and Mobile Computing**
Credit: 4	**LTP: 3-1-0**

Course objective:
This course will help students understand the fundamentals of wireless and mobile computing. Also, it gives insight into how technologies evolve over time and recent development in the area.

***Course outcome*:**
1. To study the specifications and functionalities of various protocols /standards of mobile networks.

2. To learn about the concepts and principles implementation of various Cells.
3. To explore both theoretical and practical issues of mobile computing.
4. To understand in details the working principles of various technologies (GSM, CDPD, CDMA etc)

Unit- I

Introduction to wireless communication systems: Evolution of Wireless /mobile radio communications, mobile radio systems around the world. **Radio communication systems:** paging systems, cordless telephone systems, cellular telephone systems; comparison of common wireless communications, trends in cellular radio and personal communication, Second generation (2G) Cellular networks, third generation (3G) wireless networks. **Mobile Communications:** Need, Requirement and History of Mobile communications.

Unit-II

Wireless networking: Properties of Wireless medium, Wireless local area network standards, technology- RF and IR wireless LAN, diffuse, quasi-diffuse and point –to-point IR wireless LAN, advantages and applications of wireless LAN. Introduction to WI-FI, Bluetooth, 3G and 4G wireless systems.

Unit-III

Basics of Mobile Technology: Brief history of Mobile computing. **Terrestrial Cellular Telephony:** Cellular concept, cell cluster, frequency reuse, mobile station (MS), Base station (BS), Mobile switching centre (MSC). Different Cellular Standards: digital cellular systems, TDMA and CDMA systems. Global system for Mobile communication (GSM): standard, GSM network, control function, calls setup, call handling, mobility management.

Unit-IV

Mobile Computing: Classifications of Mobile data networks, Cellular digital packet data (CDPD system).Mobile Internet: IP based mobile system, Architecture and working, General Packet Radio Service (GPRS). Switching and Traffic: intelligent cell concepts, intelligent network communication and Wireless in local loop. **Parameters of Mobile communication systems:** Design objectives, Co-channel Interference Reduction factor, adjacent channel Interference, Propagation Attenuation, and Fading.

Unit-V

Satellite mobile Communication: Orbital mechanics and satellite Classifications: GEO, MEO, LEO systems. **Global Satellite Communication**: Working of Gateway links and Inter-satellite Links, types of Satellite Changeover. Bandwidth Compression: FDMA, TDMA, CDMA, SPADE, DMAS. **Global Positioning System:** Basic principles of position fixing with GPS, errors in positioning fixing, GPS applications, case study on Google earth, VSAT systems.

Text/ References:

1. T.G. Palanivelu, R.N Akkeeran: Wireless and Mobile Communication; PHI
2. Kaveh Pahlavan, Prashant Krishnamurthy: Principles of Wireless Networks: A Unified approach: PH
3. Wireless Networks and Communications: William Stallings; MGH.
4. Comer, Computer Networks and Internets; PHI.
5. Pradip Kumar Guha Remote Sensing for the Beginner, EWP.

Course Code: CS-DSC-354	**Course Title: System Analysis and Design**
Credit: 4	**LTP: 3-1-0**

Course Objectives:

1. To introduce students to the whole systems development process.
2. To introduce the elicitation and initial modeling of information systems requirements regarding the identification of information problems.
3. To introduce the analysis and feasibility estimation, Risk analysis and work breakdown structure within a system development project setting.
4. To introduce SQA, SCM, System testing and system Security.

Course Outcomes:

1. Focuses on the elicitation and initial modeling of information systems requirements that enable identification of information problems and the subsequent analysis and modeling of an efficient solution to those problems.
2. Gaining knowledge in analysis and feasibility estimation, Risk analysis and work breakdown structure within a system development project setting aimed at developing an event-driven information system.
3. Understanding the concepts of SQA, SCM, System testing and system Security.

UNIT-I:

The system Concept: Introduction-Characteristics of System, Elements of a system, Types of system. Introduction to system development life cycle, consideration for candidate system, prototyping. Roles of system Analyst- The analyst/ user interface, the place of analyst in the MIS.

UNIT-II:

Introduction to system analysis: determining the user's information requirements, problem definition, Background analysis, fact-finding, fact analysis. Introduction to structured analysis, the tools of structured analysis, feasibility study; oral representation, data analysis, Cost/ Benefit analysis, the system proposal

UNIT- III:

Introduction to system designs: The process of design (logical and physical design), design methodology, structured design, structured walkthrough, major development activities, data validation. Introduction to input design, output design, forms design. File structure, file organization, database design, and the role of DBA.

UNIT- IV:

Introduction to system testing: The Test Plan, quality assurance, system conversion, post implementation review, software maintenance. Procedure for hardware/software selection, project management and control, project control, gantt chart, PERT and CPM.

UNIT-V:

Unified modeling language: Introduction, Use case modeling and UML, developing a use case modeling, The primary elements of UML- structural things, behavioral things, grouping things, Relationships-structural, behavioral, diagrams-structural, behavioral. A proven methodology, the importance of UML for modeling.

***Text Books*:**

1. Elias. M. Awad, System Analysis and Design, 2nd Ed., Galgotia Publication, 1997.
2. 1. Kendall and Kendall, System Analysis and Design, 8thEd., PHI, 2008.

***References*:**

1. Igor Hawryszkiewycz, Introduction to System Analysis and Design, 4thEd., PHI, 2000.
2. Rajib Mall, Fundamentals of Software Engineering, 3rd edition, PHI, 2009.

Course Code: CS-DSM-351 Course Title: Lab on a)Object Oriented Programming with C++

Credit :4 LTP: 0-0-8

Program Objectives:

1.To strengthen problem solving ability by using the characteristics of an object-oriented approach.
2.To design applications using object-oriented features
3.To handle Exceptions in programs.
4.To teach the student to implement object-oriented concepts

Program Outcomes:

1.Understand the features of C++ supporting object-oriented programming
2.Understand the relative merits of C++ as an object-oriented programming language
3.Understand how to produce object-oriented software using C++
4.Understand how to apply the major object-oriented concepts to implement object
5.oriented programs in C++, encapsulation, inheritance and polymorphism
6.Understand advanced features of C++ specifically stream I/O, templates and operator Overloading

List of the Programs:

1.Write a C++ program to find the sum of individual digits of a positive integer.
2.Write a C++ program to generate the first n terms of the sequence.
3.Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
4.Write a C++ program to find both the largest and smallest number in a list of integers.
5.Write a C++ program to sort a list of numbers in ascending order.
6.Write a Program to illustrate New and Delete Keywords for dynamic memory allocation
7.Write a program Illustrating Class Declarations, Definition, and Accessing Class Members.
8.Program to illustrate default constructor, parameterized constructor and copy constructors
9.Write a Program to Demonstrate the
a.Operator Overloading

b.Function Overloading.

10.Write a Program to Demonstrate Friend Function and Friend Class.

11.Write a Program to Access Members of a STUDENT Class Using a Pointer to Object Members.

12.Write a Program to Generate Fibonacci Series using a Constructor to Initialize the Data Members.

13.Write a C++ program to implement the matrix ADT using a class. The operations supported by this ADT are:

a.Reading a matrix.

b.Addition of matrices.

c.Printing a matrix.

d.Subtraction of matrices.

e.Multiplication of matrices

14.Write C++ programs that illustrate how the following forms of inheritance are supported:

a. Single inheritance
b. Multiple inheritance
c. Multi-levelinheritance
d. Hierarchical inheritance

2. Write a C++ program that illustrates the order of execution of constructors and destructors when a new class is derived from more than one base class.
3. Write a Program to Invoking Derived Class Member Through Base Class Pointer.
4. Write a Template Based Program to Sort the Given List of Elements.
5. Write a C++ program that uses function templates to find the largest and smallest number in a list of integers and to sort a list of numbers in ascending order.
6. Write a Program Containing a Possible Exception. Use a Try Block to Throw it and a Catch Block to Handle it Properly.
7. Write a Program to Demonstrate the Catching of All Exceptions.

Course Code: CS-DSM-351 **Course Title: Lab on b)Programming in JAVA**

Credit: 4 **LTP: 0-0-8**

Course Objectives:

1. To create a class and implement 1-D array, 2-D array, matrix addition, subtraction, multiplication, and division operations, demonstrate polymorphism, method overloading, default constructor, parameterized constructor, single level inheritance, multilevel inheritance using super and without super keyword

2. To implement exception handling using try, catch, throw, throws, and finally methods, create single thread, multi- thread using thread class and using runnable interface and also implement the join, is alive method along with priority of a thread.

3. To implement event handling on keyboard, mouse, button, check box, radio button, list, choice, text field using frame class.

Course outcomes:

After successful completion of the course students will be able to

1. Write a program to implement 1-D array, 2-D array, matrix addition, subtraction, multiplication, and division operations, demonstrate polymorphism, method overloading, default constructor, parameterized constructor, single level inheritance, multilevel inheritance using super and without super keyword

2. Write a program to implement exception handling using try, catch, throw, throws, and finally methods, create single thread, multi- thread using thread class and using runnable interface and also implement the join, is alive method along with priority of a thread.

3. Write a program to implement event handling on keyboard, mouse, button, check box, radio button, list, choice, text field using frame class.

4. Develop a GUI application of your choice that includes label, button, check box, radio button, list, choice, text field, text area along with JDBC connectivity with the database and also implement the event handling on desired components.

Following are some areas of JAVA for laboratory programming assignments but the assignments should not be limited to these only.

List of experiments:

1. Write a java program to implement 1-D, 2-D arrays. Implement the matrix addition, subtraction, multiplication, and division operations.

2. Design a class for Complex numbers in Java. In addition to methods for basic operations on complex numbers, provide a method to return the number of active objects created.

3. Design a Date class similar to the one provided in the java.util package.

4. Develop with suitable hierarchy, classes for Point, Shape, Rectangle, Square, Circle, Ellipse, Triangle, Polygon, etc.

5. Design a simple test application to demonstrate polymorphism, method overloading, default constructor, parameterized constructor, single level inheritance, multilevel inheritance using super and without super keyword.

6. Develop two different classes that implement this interface, one using array and the other using linked-list. Provide necessary exception handling in both the implementations.

7. Write a Java program to implement exception handling using try, catch, throw, throws, and finally

methods.

8. write a java program to create single thread, multi- thread using thread class and using runnable interface and also implement the join, isalive method along with priority of a thread.

9. Develop a simple paint-like program that can draw basic graphical primitives in different dimensions and colors. Use appropriate menu and buttons.

10. Develop a scientific calculator using event-driven programming paradigm of Java.

11. Implement event handling on button, check box, radio button, list,choice, text field using frame class.

12. Develop a GUI application of your choice that includes label, button, check box, radio button, list, choice, text field, text area along with JDBC connectivity with the database.

Semester VII

Course Code: CS-DSC-401 Course Title: Design and analysis of Computer Algorithms

Credit: 4 **LTP: 3-1-0**

Course Objectives:

1. To study the complexity and effectiveness of algorithms.
2. To understand different types of approaches to design algorithms and analyze them .
3. To study the different standard classical problems and the available algorithms to solve them.

Course Outcomes:

1. Students will be able to analyze the running time and prove the correctness of basic algorithms.
2. Will be able to design efficient algorithms for moderately difficult computational problems.
3. Will be able to learn the hardness of Non-Polynomial problems using simple reductions.
4. Students will be able to analyze the performance of simple approximation algorithms.

UNIT-I:

Definition of an Algorithm, analyzing an algorithm, asymptotic notation.Solution of recurrences: Substitution method, recursion tree method, master method.Basic data structure: Stacks and queues, priority queues, heap and heap sort.

UNIT-II:

Divide and Conquer: the general method, merge sort, quick sort, randomized version of quick sort, strassen's matrix multiplication, finding the maximum and minimum. Binary search tree: binary search, Insertion and deletion in trees.

UNIT-III:

Dynamic Programming: General methods, Rod cutting problem, Matrix-chain Multiplication, longest common subsequences, optimal binary search trees;Greedy method: Knapsack problem, Huffman Trees, Minimum spanning tree, algorithms of Kruskal and Prim.

UNIT-IV:

Graph Algorithms: Representation of Graphs, breadth first and depth first search, strongly connected components , topological sort. Shortest paths: Bellman-Ford Algorithm, Dijkstra's algorithm, Warshall and Floyd Algorithms. Maximum flow problem: Bipartite Graphs, Maximum matching in Bipartite Graph.

UNIT-V:

Computational Geometry algorithms: Line segment properties , finding the convex hull , finding the closest pair of points.NP Completeness: Polynomial time verification, NP Completeness and reducibility, NP Complete and NP hard problems; Approximation algorithms.

Text Books/References:

1. Introduction to Algorithms , Thomas H Cormen et al , PHI

2. Fundamentals of Computer Algorithms , Ellis Horowitz , SartajSahni , Rajasekaran , Universities Press
3. Introduction to Design and analysis of Algorithms , AnanyLevitin , Pearson
4. Algorithm Design , Jon Kleinberg , Eva Tardos , Pearson
5. Fundamentals of Algorithmics , Brassard and Bratley , PHI

Course Code: CS-DSC-402 **Course Title: Theory of Computation**

Credit: 4 **LTP: 3-1-0**

Course Objective & Outcomes :
1.Acquire knowledge about the fundamental concepts Theoretical Computer Science
2.Understand the concept of Finite Automata, Nondeterministic Finite Automata and Pushdown Automata.
3.Advance knowledge about Regular Languages and Context Free Grammar
4.Attain knowledge about Turning Machine

UNIT-I:
Sets, Relations and Functions, Fundamental Proof Techniques, Alphabets, Strings and languages, Finite and Infinite sets, Finite Representation of Languages, Regular Expressions, Deterministic and Nondeterministic Finite Automata (DFA and NFA), Equivalence of DFA and NFA.

UNIT-II:
Properties of the languages Accepted by Finite Automata, State Minimization of a DFA, Pumping Lemma for Regular Sets, Regular and Non-regular languages.

UNIT-III:
Context-free Grammars, Parse Trees, Regular Language and context free language, Chomsky's Normal Form, Pushdown Automata, Properties of Context Free Languages, Pumping Lemma for Context Free Languages, Determinism and Parsing.

UNIT-IV:
The definition of a Turing Machine, Computing with TM, Recursive and Recursively Enumerable Language, Extensions of Turing Machines, Non Deterministic Turing Machines, Chomsky's Hierarchy.

UNIT-V:
Primitive and n-Recursive Function, Church's thesis, The Halting problem, Unsolvability, Computational Complexity.

Text books/references:

1. H.R.Lewis & C.H. Papadimitriou : Elements of The Theory of Computation, P.H.I.
2. J.E.Hopcroft, R.Motwani and Ullman : Introduction To Automata Theory, Language and Computation, Pearson Education

3. K.L.P.Mishra, N.Chandrasekaran: Theory of Computer Science(Automata, Languages And Computation) ,PHI
4. John Martin: Introduction to languages and Theory of Computation, McGraw Hill
5. D.A.Cohen : Introduction To Computer Theory (J.Wiley).

Course Code: CS-DSC-403 **Course Title: Artificial Intelligence**

Credit: 4 **LTP: 3-1-0**

Course Objectives:

1. To introduce the historical evolution of Artificial Intelligence (AI), its foundations and applications.
2. To introduce the basic principles, techniques, and applications of AI and machine Learning (ML).
3. Experiment implementation with an AI programming language (Prolog/Python).
4. Explore the scope, potential, limitations, and implications of intelligent systems.

Course Outcomes:

Upon successful completion of this course, the student shall be able to:

1. Demonstrate fundamental understanding of the history of Artificial Intelligence (AI), its foundations and applications.
2. Apply basic principles of AI techniques in solutions that require problem solving, inference, perception, knowledge representation, and learning.
3. Formalize and design solutions to practical problems using the AI and ML strategies introduced during the course.
4. Demonstrate proficiency developing applications in an AI programming language (Prolog/Python).

UNIT-I:

Introduction : Foundations of Artificial Intelligence (AI), History of AI, State of the Art, Risks and Benefits of AI, Intelligent Agents, Brief discussion of applications of AI (Expert System, Natural Language Processing, Speech and Pattern Recognition etc.), Problems, Problem Spaces, and Search: Defining the Problem as a State Space Search, Production Systems, Control Strategies, Problem Characteristics, Issues in the Designing of Search Programs.

UNIT-II:

Problem-solving: Solving Problems by Searching - Problem-Solving Agents, Example Problems, Search Algorithms, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions, Search in Complex Environments, Adversarial Search and Games, Constraint Satisfaction Problems.

UNIT-III:

Knowledge, reasoning, and planning: Logical Agents, First-Order Logic, Inference in First-Order Logic, Knowledge Representation, Automated Planning, Uncertain knowledge and reasoning - Quantifying Uncertainty, Probabilistic Reasoning.

UNIT-IV:

Machine Learning: Forms of learning - Supervised Learning, Unsupervised Learning, Reinforcement Learning; Learning Probabilistic Models, Deep Learning

UNIT-V:

Application areas and AI languages: Computer Vision, Natural Language Processing, Robotics, etc.; Philosophy, Ethics, and Safety of AI ,AI languages and their important characteristics - Prolog/Python

Text Books/References:

1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, 4th Edition, Pearson Education
2. Artificial Intelligence, E. Rich, K. Knight, S. B. Nair, 3rd Edition, McGraw Hill Education
3. Artificial Intelligence, P. H. Winston, 3rd Edition, Pearson
4. Introduction to Artificial Intelligence, E. Charniak & D. McDermott, A. Wesley
5. PROLOG Programming Techniques and Application, S. Garavaglia, Harper and Row
6. Introduction to Turbo PROLOG, Carl Townsend, Sybex Inc., U.S.
7. PROLOG: Programming for Artificial Intelligence, BRATKO, 3rd Edition,Pearson Education India
8. The Complete reference Python, Martin C. Brown, 4th Edition, McGraw-Hill
9. Problem Solving and Python Programming, E. Balagurusamy, McGraw-Hill
10. Core Python Programming, Dr. R. Nageswara Rao, 3rd Edition, Dreamtech

Course Code: CS-DSC-404 **Course Title: Machine Learning**

Credits: 4 **LTP: 3 – 1 – 0**

Course Objective:

1. To explain the basics of machine learning and its applications.
2. To explain the use of knowledge in learning.
3. To explain the statistical methods in machine learning.
4. To explain the reinforcement learning and clustering of data.
5. To explain the machine learning techniques.

Course Outcome:

Upon completion of this course, the students will be able to:

1. Design machine learning models and implement them.
2. Perform the computation parts of any research oriented problems with respect to machine learning.
3. Conceive the idea to start Python programs to solve basic level AI and ML problems.
4. Create and execute basic ML programs.
5. Extend the application of ML for other areas.

UNIT – I

OVERVIEW OF MACHINE LEARNING : Introduction to machine learning, forms of learning, learning from observations, inductive learning, concept learning and the general-to-specific ordering, version spaces and the candidate-elimination algorithm, learning decision trees and ensemble learning, computational learning theory.

UNIT – II

KNOWLEDGE IN LEARNING & REGRESSION: Logical formulation of learning, knowledge in learning, explanation-based learning (EBL), relevance-based learning (RBL), knowledge-based inductive learning (KBIL), inductive logic programming (ILP)**,** Linear Regression, Multivariate Regression, Linear Classification, Logistic Regression.

UNIT – III

STATISTICAL LEARNING METHODS: Statistical learning, probability theory, learning with complete data, discrete models: Naïve Bayes models, continuous models: parameter learning, learning BayesNet structure, learning with hidden variables: expectation-maximization algorithm, Unsupervised clustering, overview of support vector machines (SVM).

UNIT – IV

REINFORCEMENT LEARNING and CLUSTERING: Introduction to Reinforcement learning, Use of Reinforcement learning, Advantages & Disadvantages, passive reinforcement learning, active reinforcement learning, generalization in reinforcement learning, overview of Clustering.

UNIT – V

APPLICATIONS OF MACHINE LEARNING: Use of ML in various fields: image processing, natural language processing, speech processing, data mining, overview of future ML, concept of deep learning.

Text books:

1. Tom M. Mitchell, Machine Learning, McGraw Hill Education, Indian Edition, 2017
2. S Sridhar and M Vijayalakshmi, Machine Learning, Oxford University Press, 2021

Reference Books:

1. Anuradha Srinivasaraghavan and Vincy Joseph, Machine Learning, Wiley Press, 2019
2. Manaranjan Pradhan and U Dinesh Kumar, Machine Learning using Python, IIMB Edition, Wiley Press, 2019
3. Sujit Bhattacharyya and Subhrajit Bhattacharyya, Practical Handbook of Machine Learning, GK Publications, 2021

E – BOOKS

3. https://alex.smola.org/drafts/thebook.pdf
4. https://ai.stanford.edu/~nilsson/MLBOOK.pdf

MOOCs

3. https://onlinecourses.nptel.ac.in/noc23_cs18
4. https://towardsdatascience.com/

Course Code: CS-DSM-401 **Course Title:Lab on a) Design and Analysis of Computer Algorithms**

Credits: 4 **LTP: 0 – 0 – 8**

Course Objectives:

The objective of laboratory paper on Design and Analysis of Computer Algorithms is to learn implementation of different algorithms to understand the running time complexity practically. Students will execute different algorithms of incremental methods, divide and conquer methods, dynamic programming, backtracking, heuristic approach and evaluate the effectiveness.

Course Outcomes:

1. Students will be able to implement different data structures and analyze the running time.
2. Students will be able to understand the effectiveness of divide and conquer methods by implementing different standard algorithms.
3. Students will be able to compare the performance of Dynamic programming with the divide and conquer method.
4. Students will be able to design different algorithms for NP-HARD and NP complete problems.

Problems related to Design and Analysis of Computer Algorithms should be solved by using the Programming languages C/C++/JAVA (preferably on Unix/Linux/Solaris operating systems environment on a network).

Following are some areas of Design and Analysis of Computer Algorithms for laboratory programming assignments but the assignments should not be limited to these only.
1. Stack and queues, tree, heap and heap sort, graphs and hashing.
2. Divide and conquer method: binary search, merge sort, quick sort, matrix multiplication, minimum spanning tree.
3. Dynamic programming: multistage graphs, all pair shortest paths, optimal binary search trees/I knapsack, traveling salesperson problem, flow shop scheduling.
4. Search and traversal techniques: AND/OR graphs, game trees, bi connected components and depth search.
5. Backtracking: Hamilton cycles, the fast Fourier transform, NP-HARD and NP complete problems.

Course Code: CS-DSM-401 **Course Title: Lab on b) Artificial Intelligence**

Credits: 4 **LTP: 0 – 0 – 8**

Lab on Artificial Intelligence Problems should be solved by using the programming language Prolog/Python

1. Write a Program to Implement Breadth First Search.
2. Write a Program to Implement Depth First Search
3. Write a Program for Pre-order, In-order and Postorder traversal of binary trees
4. Write a program to implement Hill Climbing Algorithm
5. Write a program to implement A* Algorithm
6. Write a Program to Implement Tic-Tac-Toe game
7. Write a Program to Implement 8-Puzzle problem
8. Write a Program to Implement Tower of Hanoi
9. Write a Program to Implement Water-Jug problem
10. Write a Program to check whether a given list is palindrome or not?
11. Write a Program to Implement Traveling Salesman Problem
12. Write a Program to Implement Monkey Banana Problem
13. Write a Program to Implement Missionaries-Cannibals Problems
14. Write a Program to Implement 4-Queens Problem
15. Write a Program for Quick Sort using cut in Prolog.
16. Given the following facts:

    Steve only likes easy courses.
    Science courses are hard.
    All the courses in the basket weaving department are easy.
    BK301 is a basket weaving course.

    Write a Program to find "What course would Steve like?"

17. Given the following facts:

    Sam likes all Indian mild food.
    Sam likes all Chinese food.
    Sam likes all Italian food.
    Sam likes chips.
    Curry, dal, tandoori, kurma are Indian food.
    Dal, tandoori, kurma are Indian mild food.
    chowmein, chopsuey, sweet and sour are Chinese food.
    Pizza and spaghetti are Italian food.

    Write a Program to find a) What foods does Sam like?, b) Does Sam like Curry?, c) Does Sam like Chips?

18. Given the following facts:

    All people who are not poor and are smart are happy.
    Those people who read are not stupid but smart.
    John could read and is wealthy.
    Happy people have exciting lives.

Write a Program to prove “John has an exciting life”.

19. Given the following facts:

    John likes all food.
    Apples are food.
    Chicken is food.
    Anything anyone eats & isn’t killed by is food.
    Bill eats Peanuts and is still alive.
    Sue eats everything Bill eats.

    Write a Program to prove that “John likes peanuts” and find “what food does Sue eat?”

20. Given the following facts:

    Marcus was a man.
    Marcus was a Pompeian.
    All Pompeian’s were Roman.
    Caesar was a ruler.
    All Roman were either loyal to Caesar or hated him.
    Everyone is loyal to someone.
    People only try to assassinate ruler they are not loyal to.
    Marcus try to assassinate Caesar.
    All men are people.

WAP to find “Is Marcus loyal to Caesar?”, “Does Marcus hate Caesar?”

Semester VIII

Course Code: CS-DSC-451 **Course Title: Principles of Compiler Design**

Credits: 4 **LTP: 3 – 1 – 0**

Course Objectives:

1. A brief introduction to compiler basics, including lexical analysis and syntax trees.
2. To introduce the Top-Down and Bottom-Up parsing algorithms.
3. An introduction to the symbol table and run-time storage management.
4. Describe how Intermediate Code Generation works.
5. An explanation of code generation and code optimization.

Course Outcomes:

1. Understanding lexical analysis and syntax trees is essential for understanding compiler basics.
2. Understanding top-down and bottom-up parsing algorithms.

3. An understanding of the symbol table and how run-time storage is handled.
4. Getting a better understanding of intermediate code generation.
5. Understanding the generation and optimization of code.

UNIT-I:

Overview of compiling process, some typical compiler structures, Regular expressions, The role of lexical analyzer, Input Buffering, Specification of tokens, recognition of tokens, Syntax trees, ambiguity, Context free Grammar and derivation of parse trees.

UNIT-II:

Top-down and bottom-up parsing, operator precedence parsing, syntax-directed translation, translation schemes.

UNIT-III:

Symbol Table: The contents of a symbol table, Data structures for symbol tables
(ST), design of an ST, ST for block-structured languages Run time storage administration: Storage allocation strategies, static, dynamic and heap memory allocation, memory allocation in block structured languages, memory allocation in recursion.

UNIT-IV:

Intermediate Code Generation: Intermediate Languages, Intermediate Representation Techniques, Statements in three-address code, Implementation of three-address instruction, Three-address code generation, Code Generation for arrays, Translation of Boolean expressions, Translation of Control flow statements, Translation of Case Statements, Function Calls.

UNIT-V:

Code Generation: Basic block, Representation of Basic blocks, Problems in Code Generation, Code Generation for DAGs, Register Allocation and Register Interference Graph, Code Generation using Dynamic Programming. Code Optimization: Principal sources of optimization, loop optimization, Global data flow analysis.

Text books/References books:

1. Compilers - Principles, Techniques and tools, Alfred V Aho, R.Sethi, D. Ullman (Pearson)
2. Compiler Design - K. Muneeswaran, Oxford University Press
3. The theory and practice of compiler writing, Trembley and Sorenson, (McGraw Hill)
4. Compiler Design, Santanu Chattopadhyay, PHI

Course Code: CS-DSM-451 Course Title: Data Science and Research Methodology

Credits: 4 **LTP: 3 – 1 – 0**

Course Objective:

1. To explain the basic research, Quantitative methods, and IPR etc.
2. To explain the use of matrices and arrays in R.
3. To explain the use of non numeric values in vectors.

4. To explain the use of data frame and list in R.
5. To explain the graph plotting using ggplot2.

Course Outcome:

Upon completion of this course, the students will be able to:

1. Understand the research methodology basics concepts.
2. Identify and execute basic syntax and programs in R. Perform the Matrix operations using R built in functions.
3. Apply non numeric values in vectors.
4. Create the list and data frames.
5. Exploit the graph using ggplot2.

UNIT – I

OVERVIEW OF RESEARCH METHODOLOGY: Research: Meaning, objectives, types, approaches. Criteria of good research, research problems, research design. Review of literature: Meaning, objectives, principles and procedure. Report writing: Meaning, significance, types, techniques, essentials of writing scientific article. Quantitative methods of research: Methods of data collection – experimental data, field data, data from secondary sources. Relation between variables: correlation (both continuous &amp; binary data), regression (both linear &amp; non-linear) for two variables. Test of significance including one-way- ANOVA. Errors and analysis of errors. Intellectual Property Rights (IPR) – concept and definition, types - patents, trademarks, copyrights and trade secrets. Role of different national level GoI Funding Agency (CSIR, DST, DBT, DAE, DRDO, DOS etc.) for promotion of scientific research.

UNIT – II

OVERVIEW OF R & MATRICES AND ARRAYS: History and Overview of R- Basic Features of R-Design of the R System- Installation of R- Console and Editor Panes- Comments- Installing and Loading R Packages- Help Files and Function Documentation Saving Work and Exiting R- Conventions- R for Basic Math- Arithmetic-Logarithms and Exponentials E-Notation- Assigning Objects- Vectors- Creating a Vector- Sequences, Repetition, Sorting, and Lengths- Sub setting and Element Extraction- Vector-Oriented behavior. Defining a Matrix – Defining a Matrix- Filling Direction- Row and Column Bindings- Matrix Dimensions Sub setting- Row, Column, and Diagonal Extractions- Omitting and Overwriting- Matrix Operations and Algebra- Matrix Transpose- Identity Matrix- Matrix Addition and Subtraction- Matrix Multiplication Matrix Inversion-Multidimensional Arrays-Subsets, Extractions, and Replacements.

UNIT – III

NON-NUMERIC VALUES: Logical Values- Relational Operators- Characters- Creating a String-Concatenation- Escape Sequences Substrings and Matching- Factors- Identifying Categories- Defining and Ordering Levels- Combining and Cutting.

UNIT – IV

LISTS AND DATA FRAMES: Lists of Objects-Component Access-Naming-Nesting-Data Frames-Adding Data Columns and Combining Data Frames-Logical Record Subsets-Some Special Values-Infinity-NaN-NA-NULL Attributes-Object-Class-Is-Dot Object-Checking Functions-As-Dot Coercion Functions.

UNIT – IV

BASIC PLOTTING : Using plot with Coordinate Vectors-Graphical Parameters-Automatic Plot Types-Title and Axis Labels Color-Line and Point Appearances-Plotting Region Limits-Adding Points, Lines, and Text to an Existing Plot-ggplot2 Package-Quick Plot with ggplot-Setting Appearance Constants with Geoms-- READING AND WRITING FILES- R-Ready Data Sets- Contributed Data Sets- Reading in External Data Files- Writing Out Data Files and Plots- Ad Hoc Object Read/Write Operations.

Text Books:

1. Tilman M.Davies,"THE BOOK OF R - A FIRST PROGRAMMING AND STATISTICS" Library of Congress Cataloging-in-Publication Data,2016.

Reference Books:

1. Roger D. Peng,"R Programming for Data Science"Lean Publishing, 2016.
2. Hadley Wickham, Garrett Grolemund," R for Data Science", O'REILLY Publication,2017
3. Steven Keller, "R Programming for Beginners", CreateSpace Independent Publishing Platform 2016.
4. 4. KunRen ,"Learning R Programming", Packt Publishing,2016

E – Books:

1. https://web.itu.edu.tr/~tokerem/The_Book_of_R.pdf

MOOC

1. https://online-learning.harvard.edu/subject/r
2. https://www.udemy.com/course/r-basics/
3. https://www.datacamp.com/courses/free-introduction-to-r

Course Code: CS-DSC-452 **Course Title: Research Project/ Dissertation**

Credits: 12 **LTP: 0 – 0 – 24**