Department of Information Science and Engg.

NEP 2022 BATCH

3rd SEM SCHEME and SYLLABUS

A.Y: 2023-24

Vision of the College:
To develop thoughtful and creative young minds in a learning environment of high academic ambience by synergising spiritual values and technological competence.

Mission of the College:
1. To continuously strive for the total development of students by educating them in state-of-the-art-technologies and managerial competencies providing best in class learning experience with emphasis on skills, values and learning outcomes and helping them imbibe professional ethics and societal commitment.
2. To create research ambience that promotes interdisciplinary research catering to the needs of industry and society.
3. To collaborate with premier academic and research institutions and industries to strengthen multidisciplinary education, applied research, innovation, entrepreneurship and consulting ecosystems.

Vision of the Department:
To be a centre for quality education and research in Information Science and Engineering to create high quality professionals for catering to the need of the society.

Mission of the Department:
1) To enable students to acquire strong fundamental concepts related to the Information Science and Engineering through experiential learning.
2) To educate students towards state-of-the-art-technologies and multidisciplinary practices for a successful career by creating learning-teaching-learning ambience.
3) To inculcate life-long learning through innovation and research attitudes among students related to Information Science and Engineering.

Program Educational Objectives (PEOs):

The objectives of Information Science and Engineering degree program are to prepare students to meet the academic excellence, professionalism, and ability to solve a broad range of problems in rapidly changing technological, economic and social environment.

Graduates of the program will:
1. Pursue career as software engineer, project manager, data scientist, entrepreneur and pursue higher studies and research in Information Science and Engineering domains.
2. Apply mathematical, scientific and Information Science and Engineering knowledge with multidisciplinary approaches to solve real world problems.

3. Possess professionalism, ethical and societal responsibilities and engage in life-long learning through pursuit of skill development and certification courses in Information Science and Engineering.

Programme Outcomes (POs):

To achieve the above objectives, Information Science and Engineering degree programme strives to obtain the following outcomes which should be achieved by all graduates at the time of their graduation.

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Specific Outcomes (PSOs):

1) **Computing System:** Demonstrate the knowledge of evolving hardware and/or software to develop solutions to real life computational problems with a focus on performance optimization.

2) **Communication and Security:** Design and develop solutions for providing efficient transmission, storage, security and privacy of data in diverse computing environment.

3) **Information management:** Apply tools and techniques for management of information system, data analysis and knowledge discovery in the process of decision making.

SCHEME OF TEACHING AND EXAMINATION (2022 Scheme)

III Semester

Sl. No.	Course and Course Code		Course Title	Teaching / Paper setting Dept.	Teaching hrs./week				Examination				Credits
					Lecture L	Tutorial T	Practical/ Drawing P	Self-Study Component S	Duration in hrs.	CIE Marks	SEE Marks	Total Marks	
1.	PCC / BSC	S3MAT1	Statistics and Probability	Maths	3	0	0		3	50	50	100	3
2.	IPCC	S3ISI01	Digital Circuits and Computer Organization(Integrated)	IS	3	0	2		3	50	50	100	4
3.	IPCC	S3ISI02	Advanced Web Technology and Internet Applications(Integrated)	IS	3	0	2		3	50	50	100	4
4.	PCC	S3IS01	Data Structures	IS	3	0	0		3	50	50	100	3
5.	PCCL	S3ISL01	Data Structures Laboratory	IS	0	0	2		3	50	50	100	1
6.	ESC	S3ISSCXX	ESC/ETC/PLC	IS	3	0	0		3	50	50	100	3
7.	UHV	SHS01	Social Connect and Responsibility (Board: ME)	ME	0	0	2		-	100	-	100	1
8.	AEC/ SEC	S3ISAXX	Ability Enhancement Course/ Skill Enhancement Course – III	IS	If offered as Theory Course				1½	50	50	100	1
					1	0	0						
					If offered as Integrated Course				1½				
					0	0	2						
9.	NCMC	SMC01	National Service Scheme (NSS)	NSS CO	0	0	2			100	-	100	0
		SMC02	Physical Education (PE) (Sports and Athletics)	PED									
		SMC03	Yoga	PED									
			Total							550	350	900	20
		AAP	AICTE Activity Points (Applicable for both Regular and Lateral Entry students)	40 hours community service to be documented and produced for the examination									

Note: PCC: Professional Core Course, IPCC: Integrated Professional Core Course, PCCL: Professional Core Course laboratory, UHV: Universal Human Value Course, NCMC: Non Credit Mandatory Course, AEC: Ability Enhancement Course, SEC: Skill Enhancement Course, ESC: Engineering Science Course, ETC: Emerging Technology Course, PLC: Programming Language Course
L: Lecture, T: Tutorial, P: Practical S= SDA: Skill Development Activity, CIE: Continuous Internal Evaluation, SEE: Semester End Evaluation.

Engineering Science Course (ESC/ETC/PLC) (Offered by the Department)

S3ISES01	Object Oriented Programming with C++	S3ISES02	Multimedia Technologies
S3ISES03	Object Oriented Programming with Java	S3ISES04	Game Programming-2D

Ability Enhancement Course – III (Offered by the Department)

S3ISA01	Introduction to Blender Tool: Animation and 3D Creation	S3ISA02	Documentation: MS Office and LaTex

S3ISA03	Unix and Shell Programming		

Integrated Professional Core Course (IPCC): Refers to Professional Core Course Theory Integrated with practical of the same course. Credit for IPCC can be 04 and its Teaching–Learning hours (L : T : P) can be considered as (3 : 0 : 2) or (2 : 2 : 2). The theory part of the IPCC shall be evaluated both by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper. For more details, the regulation governing the Degree of Bachelor of Engineering (B.E.) 2022-23 may please be referred.

National Service Scheme /Physical Education/Yoga: All students have to register for any one of the courses namely National Service Scheme (NSS), Physical Education (PE)(Sports and Athletics), and Yoga(YOG) with the concerned coordinator of the course during the first Week of III semesters. Activities shall be carried out between III semester to the VI semester (for 4 semesters). Successful completion of the registered course and requisite CIE score is mandatory for the award of the Degree. The events shall be appropriately scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities. These courses shall not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the course is mandatory for the award of Degree.

Statistics and Probability

Contact Hours/ Week	3 (L)	**Credits**	3
Total Lecture Hours	40	**CIE Marks**	50
Total Tutorial Hours	00	**SEE Marks**	50
Sub Code	S3MAT1	**Semester**	III

Prerequisites: Engineering Mathematics-I and Engineering Mathematics-II.

Course objectives:

1. To develop and conduct appropriate experimentation, analyze and interpret data and use engineering judgment to draw conclusion.
2. To introduce the basic concepts and applications of probability in engineering.
3. To provide the knowledge about the random variable, random process and how to model the random processes in engineering.
4. To deal with multiple random variables and introduction of the most important types of stochastic processes.
5. To investigate the variability in sample statistics from sample to sample, measure of central tendency & dispersion of sample statistics and pattern of variability of sample.

UNIT-I

Statistics: Introduction, Definitions, Curve Fitting: Straight line, parabola and exponential curves. Correlation and regression, formula for correlation coefficient, regression lines and angle between the regression lines.

8Hours

UNIT-II

Probability: Basic terminology, Definition of probability, Probability and set notations, Addition law of probability, independent events, conditional probability, multiplication law of probability, Baye's theorem.

8 Hours

UNIT – III

Random Variable: Discrete Probability distribution, Continuous Probability distribution, expectation, Variance, Binomial distribution, Poisson distribution, Normal distribution and Exponential distributions.

8 Hours

UNIT-IV

Joint Probability: Joint probability distribution, Discrete and independent random variables, Expectation, Covariance, Correlation coefficient. Probability vectors, stochastic matrices, fixed point matrices, Regular stochastic matrices,

Markov chains, Higher transition-probabilities, stationary distribution of regular Markov chains and absorbing states.

8 Hours

UNIT-V

Sampling Distribution: Introduction, Objectives, sampling distribution, testing of hypothesis, level of significance, confidence limits, simple sampling of attributes, test of significance of large samples, comparison of large samples, sampling of variables, central limit theorem, confidence limits for unknown mean, test of significance for means of two large samples, Sampling of variables – small samples, Student's t-distribution.

8 Hours

Text Book:

1. B.S.Grewal, "Higher Engineering Mathematics", 43rd edition, Khanna Publications, 2015. ISBN:978-81-7409-195-5.
2. Ramana .B.V, "Higher Engineering Mathematics", latest edition, Tata-McGraw Hill, 2016. ISBN;0-07-053516-7.

Reference Books:

1. Erwin Kreyszig, "Advanced Engineering Mathematics" , 10th edition, Wiley Publications, 2015.
2. ISBN;978-81-7409-195-5.
3. C. Ray Wylie and Louis C. Barrett, "Advanced Engineering Mathematics", 6th Edition, Tata-McGraw Hill 2005.
4. Louis A. Pipes and Lawrence R. Harvill, "Applied Mathematics for Engineers and Physicists", 3rd Edition, McGraw Hill, 2014.

Course Outcomes:

Upon completion of this course the student will be able to:

CO1:Apply least square method to fit a curve for the given data and evaluate the correlation coefficient and regression lines for the data (L3).

CO2:Analyse the nature of the events and hence determine the appropriate probabilities of the events (L3).

CO3:Classify the random variables to determine the appropriate probability distributions (L2).

CO4:Determine the joint probability distribution, its mean, variance and covariance. Calculate the transition matrix and fixed probability vector for a given Markov chain (L3).

CO5:Estimate the parameter of a population, important role of normal distribution as a sampling distribution (L2).

Digital Circuits and Computer Organization (Integrated Course)

Contact Hours/ Week:	: 3L+2P	Credits:	4
Total Lecture Hours:	: 40	CIE Marks:	50
Sub. Code:	: S3ISI01	SEE Marks:	50

Course objectives: This course will enable students to:	
1.	Learn the different techniques of simplification of Boolean expressions and design of combinational circuits.
2.	Design counters and registers for the real time applications.
3.	Apply appropriate error detection and correction techniques for reliable data transmission.
4.	Understand switching and data link control protocols.

UNIT I

Introduction to Digital System: Boolean Algebra and Boolean Functions, min-term and max-term notations. Simplification of Boolean expressions using Boolean laws and Rules, Simplification of Boolean expressions using Karnaugh map techniques. Design of combinational Logic Circuits: Design of Decoders, Multiplexers, DeMultiplexers.

8 Hours

UNIT II

Sequential Logic Circuits

Flip Flops: Introduction to Flip-Flops, Types of Flip-Flops: RS FF, SR FF, JK FF and M/S JK FF. **Counters:** Definition of Counter, Types of Counters, Design of Asynchronous Counters.

Registers: Basic Register, Shift-Register, Types of Shift Registers, Unidirectional and Bidirectional Shift Register, Johnson counter and Ring Counters.

8 Hours

UNIT III

Basic Structure of Computer: Functional Units, Basic Operational Concepts, Bus Structures, Performance - Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. Machine Instructions and Programs: Numbers, Arithmetic Operations and Characters - Number Representation, Addition of Positive Numbers, Addition and Subtraction of Signed Numbers, Overflow in Integer Arithmetic, Characters, Memory Location and Addresses - Byte Addressability.

8 Hours

UNIT IV
Addressing modes: Implementation of Variables and Constants, Indirection and Pointers, Indexing and Arrays, Relative Addressing, Additional Modes. **Basic Processing Unit:** Execution of a Complete Instruction - Branch Instructions, Multiple Bus Organization, Hard wired Control - A Complete Processor, Micro programmed Control - Microinstructions.
8 Hours

UNIT V
Arithmetic: Addition and Subtraction of Signed Numbers - Addition/Subtraction Logic Unit, Multiplication of Positive Numbers, Signed Operand Multiplication - Booth Algorithm, Fast Multiplication - Bit-Pair Recoding of Multipliers, Carry-Save Addition of Summands, Integer Division, Floating-point Numbers and Operations - IEEE Standard for Floating-Point Numbers, Arithmetic Operations on Floating-Point Numbers - Addition and Subtraction Operations.
8 Hours

TEXT BOOKS		
1	Thomas L Floyd, Digital Fundamentals	Digital Fundamentals, Pearson Publications. Ed.11, 2020.
2	Carl Hamacher, Zvonko Vranesic, SafwatZaky	Computer Organization, Tata McGraw Hill, ED.5, 2017.

REFERENCE BOOKS		
1	Donald P Leach, Albert Paul Malvino and Goutam Saha	Digital Principles and Applications, Tata McGraw – Hill, Ed.8, 2014.
2	Ronald J Tocci, Neil S Widmer, Gegory L. Moss,	Digital Systems-Principles and Applications, Pearson Education, Ed.12. 2016.

List of Problems for laboratory

1. Design and implementation of Half-adder and Full adder using minimum number of NAND gates only.
2. Design and implement the Full adder and Full subtractor using only multiplexer and other gates.

3. Design and implement the Full adder and Full subtractor using decoder and other logic gates.
4. Implement the following using 4-bit shift register IC 74LS95.
   a) Right Shift b) SISO c) PIPO d) PISO e) SIPO
   f) Left Shift g) Ring Counter h) Johnson Counter
5. Design a sequence generator to generate the given sequence using shift Register IC and other gates.
6. Design and implement 3-stage Asynchronous (mod-8) counter using MS J-K flip flops IC7476.
7. Implement UP-Down pre-settable counter using IC 74LS190 for the given mod N.
8. Design and implement a 3-bit binary mod-n synchronous counter using MS J-K FF IC74LS76.

Course Outcomes: Upon completion of this course the student will be able to:	
CO1:	**Design** combinational logic circuits for the required applications.
CO2:	**Design** sequential logic circuits such as counters and registers for the specific needs**.**
CO3:	**Analyze** the performance of a basic computer system.
CO4:	**Apply** the knowledge of addressing modes to develop assembly language code and Design control sequence for the given instruction on different CPU bus structures.
CO5:	**Apply** appropriate technique to solve arithmetic related problems in computer.

Advanced Web Technology and Internet Applications (Integrated Course)

Contact Hours/ Week:	: 3L + 2P	Credits:	4
Total Lecture Hours:	: 40	CIE Marks:	50
Sub. Code:	: S3ISI02	SEE Marks:	50

Course objectives:	
This course will enable students to:	
1.	Acquire knowledge and skills for creation of web site considering both client and server-side programming.
2.	Gain ability to develop responsive web applications.
3.	Acquire skills to validate and handle errors using PHP.
4.	Create web services using XML, JSON and PHP.

UNIT I

Introduction to Web Development: A Complicated Ecosystem, Definitions and History, The Client-Server Model, Where Is the Internet? Working in Web Development.

How the Web Works: Internet Protocols, Domain Name System, Uniform Resource Locators, Hypertext Transfer Protocol, Web Browsers, Web Servers.

8 Hours

UNIT II

Introduction to HTML: What is HTML and Where did it come from? HTML Syntax, Semantic Markup, Structure of HTML Documents, Quick Tour of HTML Elements, HTML5 Semantic Structure Elements.

Introduction to CSS: What is CSS, CSS Syntax, Location of Styles, Selectors, The Cascade: How Styles Interact, The Box Model, CSS Text Styling.

8 Hours

UNIT III

JavaScript: Client-Side Scripting: What is JavaScript and What can it do? JavaScript Design Principles, where does JavaScript Go? Syntax, JavaScript Objects, The Document Object Model (DOM), JavaScript Events, Forms.

Introduction to Server-Side Development with PHP: What is Server-Side Development, A Web Server's Responsibilities, Quick Tour of PHP, Program Control, Functions

8 Hours

UNIT IV

PHP Arrays and Superglobals: Arrays, $_GET and $_POST Super global Arrays, $_SERVER Array, $_Files Array, Reading/Writing Files.

Error Handling and Validation: What are Errors and Exceptions? PHP Error

Reporting, PHP Error and Exception Handling.
8 Hours

UNIT V
Advanced JavaScript and jQuery: JavaScript Pseudo-Classes, jQuery Foundations, AJAX, Asynchronous File Transmission, Animation, Backbone MVC Frameworks. **XML Processing and Web Services**: XML Processing, JSON, Overview of Web Services, Creating and Consuming Web Services in PHP, Interacting Asynchronously with Web Services.
8 Hours

TEXT BOOKS		
1	Randy Connolly, Ricardo Hoar	Fundamentals of Web Development, Pearson Education India, 2nd Edition, 2018, ISBN:978-9332575271.

REFERENCE BOOKS		
1	Robin Nixon	Learning PHP, MySQL &JavaScript with jQuery, CSS and HTML5", 4th Edition, O'Reilly Publications, 2015. (ISBN:978-9352130153).
3	Dean Wampler	Programming Hive, , O'Reilly, Kindle Publication.

List of Problems for laboratory:

1. To design user interface for a given scenario using basic HTML tags.
2. To demonstrate the concepts of CSS selectors and conflict resolution.
3. To demonstrate the concepts of various UI components.
4. To demonstrate the concepts of syntactic structures of JavaScript.
5. To demonstrate the Client side validation using JavaScript.
6. To construct a JSON and XML structures.
7. To demonstrate the working of Server side program with forms using PHP.
8. To demonstrate the database access with PHP.

Course Outcomes: Upon completion of this course the student will be able to:	
CO1:	**Comprehend** web development and its working.
CO2:	**Apply** HTML and CSS to build web pages.
CO3:	**Apply** client-side and server-side scripting languages to develop web applications.
CO4:	**Implement** validation and error handling using PHP to build efficient server-side web applications.
CO5:	**Apply** and **implement** JSON and XML scripts to share information across web applications.

Data Structures

Contact Hours/Week	:	3L	Credits	:	3.0
Total Lecture Hours	:	40	CIE Marks	:	50
Course Code	:	S3IS01	SEE Marks	:	50

Course objectives:
This course will enable students to:

1. Describe the efficient data storage mechanisms for easy access.
2. Describe the properties of various data structures such as stacks, queues, lists, and trees.
3. Implement Stack and Queue data structures and their applications.
4. Design and implement various types of linked lists, trees and their applications.
5. Apply the knowledge in selecting an appropriate data structure for a problem to be solved.

UNIT I

Structures and Unions: Defining a Structure, declaring Structure variables, accessing Structure members, Structure initialization, copying and comparing Structure variables, operations on individual members, array of Structures, array within Structure, Structure within Structure, Structures and Functions, Unions, size of structures.
File management in C: Defining and Opening a file, Closing a file, Input/Output operations on files - getc(), putc(), getw(), putw(), fscanf(), fprintf(), Error handling during I/O operations - feof(), ferror(), Random access to files - ftell(), rewind(), fseek(), Command line arguments.
(Text Book 1: 10, 12)

8Hours

UNIT II

The Stack: Definition and Examples, representing Stacks in C, Infix, Applications of stacks: Postfix, and Prefix expressions, conversion and evaluation.
Recursion: Recursive Definition and Processes, Recursion in C, Writing recursive programs: The Towers of Hanoi Problem, Efficiency of Recursion.
Queues: The Queue and Its Sequential Representation: C implementation of Queues, Insertion, Deletion and Display operations, Types of Queues (Linear, Circular).
(Text Book 2: 2, 3.1, 3.2, 3.3(only the Towers of Hanoi Problem), 3.5. 4.1(excluding Queue as an ADT))

8Hours

UNIT III

Dynamic memory allocation: malloc(), calloc(), realloc(), free().
(Text Book 1: 13.1-13.6)
Linked lists: Inserting and removing nodes from a list, linked implementation of stacks, getnode and freenode operations, linked implementation of queues, examples of list operation, list implementation of priority queues, header nodes.

Lists in C: allocating and freeing dynamic variables, linked lists using dynamic variables, queues as lists in C, examples of list operations in C, non-integer and non-homogeneous lists, Addition of two polynomials, implementing header nodes. (Text Book 2: 4.2, 4.3(except array implementation of list, Limitations of array implementation, comparing dynamic and array implementations of list))

8Hours

UNIT IV

Other List Structures: Circular lists, stack as a Circular list, queue as a Circular list, primitive operations on circular lists, the Josephus problem, header nodes, Doubly linked lists, Primitive operations on Doubly linked list.
(Text Book 2: 4.5(except addition of long positive integers using circular and doubly linked list))

8 Hours

UNIT V

Trees: Operations on Binary Trees, Applications of Binary Trees, Binary Tree Representations: Node representation of Binary Trees, Internal and External Nodes, Implicit array representation of Binary Trees, Binary Tree Traversals in C.
Trees and Their applications: C Representations of Trees, Tree Traversals, General Expressions as Trees, Evaluating an Expression Tree, Constructing a Tree.
(Text Book 2: 5.1, 5.2, 5.5(except choosing Binary Tree Representation, Traversal using a Father field, Heterogeneous Binary Trees))

8Hours

Self-study Concepts:
Priority and Double Ended Queues
Threaded Binary Trees - definition and types

NOTE: Self-study topics are assessed only in CIE-Quizzes/ Assignments

TEXT BOOKS:

1.	E.Balagurusamy	Programming in ANSI C, Edition 7, Tata McGraw-Hill Publications, 2017. (Unit I)
2.	Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum	Data structures using C and C++, PHI/Pearson, Edition 2, 2015. (Unit II to V)

REFERENCE BOOKS:

1.	Horowitz, Sahni and Anderson-Freed	Fundamentals of Data Structures in C, Edition 2, Universities Press Pvt. Ltd., 2011

Course outcomes:
After the completion of this course, students will be able to:

CO1: Apply advanced C programming techniques like pointers, structures and files to develop solutions for given problems.

CO2: Implement different data structures like Stacks and Queues using static memory allocation technique.

CO3: Implement different types of Linked Lists using dynamic memory allocation technique.

CO4: Apply the knowledge of Stacks, Queues and linked lists to design and develop solutions to given problems.

CO5: Implement non-linear data structures such as trees and their applications using dynamic memory allocation technique.

Data Structures Laboratory

Contact Hours/Week	:	2	Credits	:	1.0
Total Lecture Hours	:	-	CIE Marks	:	50
Total Practical Hours	:	26	SEE Marks	:	50
Course Code	:	S3ISL01			

Course Objectives:

This course enables the students to

1. Develop and implement Linear data structures and their applications such as stacks, queues using static memory allocation.
2. Develop and implement Linear data structures such as linked lists using dynamic memory allocation.
3. Explore the applications of linked lists, develop and implement them.
4. Develop and implement Non-Linear data structures such as trees and their applications.

List of Programs

1. Write a C program to create a sequential file with at least five records, each record having the structure shown below:

EMPLOYEE_ID	NAME	DEPARTMENT	SALARY	AGE
Non-Zero Positive integer	25 Characters	25 Characters	Positive Integer	Positive integer

Write necessary functions to perform the following operations:
a) to display all the records in the file.
b) to search for a specific record based on EMPLOYEE_ID.

2. Develop and implement a STACK of integers using array and perform the following operations: (a) PUSH (b) POP (c) DISPLAY and (d)check whether the contents of stack form a palindrome.

3. Write a C program to convert
(a) the given infix expression to postfix expression.
(b) the given infix expression to prefix expression.
(c) to evaluate a given prefix/postfix expression.

4. Develop and implement linear QUEUE of strings using array and perform the following operations: (a) insertion, (b) deletion and (c) display.

5. Develop and implement CIRCULAR QUEUE of integers using array and perform the following operations: (a) insertion, (b) deletion and (c) display.

6. Develop and implement singly linked list with integer data and perform the following operations:
a) to insert a node at the end of the list.

b) to delete the first node in the list.
c) to insert a node at the specified position in the list (1**<=pos<=n** where **'n'** is the total number of nodes in the list & **'pos'** is the position where data is to be inserted).
d) to display the contents of the list.
e) to reverse a given list.

7. Develop and implement two ***ordered singly linked lists*** with the following operations:
   a) insert into list1.
   b) insert into list2.
   c) to perform UNION of list1 and list2
   d) to perform INTERSECTION of list1 and list2
   e) display the contents of all three lists.

8. Develop and implement a STACK of integers using singly linked list and perform the following operations: (a) PUSH (b) POP (c) DISPLAY.

9. Develop and implement linear QUEUE of integers using singly linked list and perform the following operations: (a) insertion, (b) deletion and (c) display.

10. Develop and implement addition of two polynomials with two coefficients using singly linked lists.

11. Develop and implement doubly linked list with header node with the following operations: (Header node should maintain the count of number of nodes in the list after each operation).
    a) Insert a node at the end of the list.
    b) Insert a new node next to a node whose information field is specified.
    c) To delete first node if pointer to the last node is given.
    d) To delete a node whose information is given.
    e) To display the contents of the list.
    f) To swap n^{th} and m^{th} nodes in the list.

12. Develop and implement DEQUE using doubly linked list to perform the following operations: insertion, deletion and display.

13. Develop and implement binary search tree (BST) of integers to perform the following operations:
    a) Insert into a BST.
    b) Traverse the tree in inorder/ preorder/ postorder.
    c) Delete a given node from the BST.

14. Develop and implement an expression tree for a given valid postfix expression and evaluate the expression tree.

Course Outcomes (COs):

Upon successful completion of the course the student will be able to:

CO1: **Design** and **develop** C programs by applying C programming techniques like pointers, structures and files to develop solutions for particular problems.

CO2: **Design** and **develop** Linear data structures like Stack, Queue using static memory allocation technique and explore their applications.

CO3: **Design** and **develop** Linear data structures like Linked Lists using dynamic memory allocation technique.

CO4: **Apply** the knowledge of linked lists to **design** and **develop** solutions to given problems.

CO5: **Apply** the knowledge of dynamic memory allocation technique to develop and implement non-linear data structures like Trees and their applications.

Object Oriented Programming with C++
(Engineering Science Course-III-1)

Contact Hours/ Week:	: 3	Credits:	3
Total Lecture Hours:	: 40	CIE Marks:	50
Sub. Code:	: S3ISES01	SEE Marks:	50

Course objectives:
This course will enable students to:

1. Understand the basic principles of object-oriented programming using C++.
2. Analyze important oops concepts such as classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, templates.
3. Discuss how Object-oriented programming is used in modular designing and also as a reusable software system.

UNIT I

Principles of Object-Oriented Programming and Functions: Procedure oriented Programming, Object oriented Programming Paradigm, basic concepts of object oriented programming, Benefits of OOPS, Object oriented languages, Applications of OOP, Reference Variables, Operators in C++, Scope resolution Operator, Memory dereferencing operators Memory management operators, Manipulators, Function prototyping, call by reference, return by reference, inline functions, default arguments, const arguments, function overloading.

Classes and Objects: C structure revisited, specifying a class, Defining member functions, A C++ program with class static member functions, Making an outside function inline, Nesting of member functions, Private member functions, Arrays within a class, memory allocation for objects, static data members, Array of objects, objects as function arguments, Friend functions, returning objects.

9 Hours

UNIT II

Constructors and Destructors : Constructors, parameterized constructors, Copy constructors, Multiple constructors in a class, Constructors with default arguments, Dynamic initialization of objects, Dynamic constructors

Operator Overloading and Type Conversions: Introduction, Defining operator overloading, Overloading Unary and Binary operators, Overloading binary operators using friend function, Manipulation of strings using operators, Rules for overloading operators, type conversions.

8 Hours

UNIT III

Inheritance: Introduction, Defining derived classes, Single inheritance, making a private member inheritable, Multi-level inheritance, Multiple inheritance, Hierarchical inheritance, Hybrid inheritance, Virtual base

classes, Abstract classes, Constructors in derived classes, Nesting of classes. **Virtual Functions and Polymorphism**: Introduction, Pointers, pointers to objects, this pointer, pointers to derived classes Virtual functions, pure virtual functions.
8 Hours

UNIT IV
Templates: Introduction, class templates, Class templates with multiple parameters, Function templates, Function templates with multiple parameters, Overloading of template functions, Member function templates, non-type template arguments.
8 Hours

UNIT V
Exception Handling: Introduction, basics of Exception handling, Exception handling mechanism, Throwing mechanism, catching mechanism, rethrowing an exception, specifying exceptions, Exceptions in constructors and destructors, exceptions in operator overloaded functions.
7 Hours

TEXT BOOKS		
1	E. Balagurusamy	Object Oriented Programming with C++, 6th edition, TMH, 2013, ISBN: 9781259029936, 125902993X.

REFERENCE BOOKS		
1	Herbert Schildt	The Complete Reference C++, 5th Edition, TMH, 2012
2	Stephen R. Davis	C++ for Dummies, 7th Edition, John Wiley and Sons Inc, 2014

Course Outcomes: Upon completion of this course the student will be able to:	
CO1.	Apply various C++ constructs such as classes, functions, function overloading and dynamic memory management to develop solutions to problems.
CO2.	Develop programs using constructors, destructors, illustrate Operator overloading and type conversion concepts.
CO3.	Design programs using Inheritance to achieve code reusability and virtual functions to achieve run time polymorphism.
CO4.	Design and apply Templates to handle real world problems.
CO5.	Analyze different exceptions and develop programs to handle various exceptions.

Multimedia Technologies
(Engineering Science Course III-2)

Contact Hours/ Week:	3	Credits:	3
Total Lecture Hours:	40	CIE Marks:	50
Sub. Code:	S3ISES02	SEE Marks:	50

Course objectives:
This course will enable students to:

1.	Explain the definition of Multi-media and Hypermedia and apply the concepts of digitization of audio.
2.	Apply and Analyze text data using lossless compression algorithms.
3.	Apply lossy compression algorithms and transform coding and wave-let based coding for two-dimensional data.
4.	Compare JPEG and JPEG 2000 Standards.
5.	Outline overview of MPEG-1 and MPEG-2. Interpret the MPEG video coding- II for MPEG-4 & 7.

UNIT I

Introduction to Multimedia, Graphics and Image Representation, Fundamental concepts in Video and audio: Definition of Multimedia, Multi-media and Hypermedia. Graphics/Image Data Types, Popular File Formats. Types of Video Signals, Analog Video, Digital Video, Digitization of sound, Quantization and Transmission of Audio.
Text: Chapter 1, 1.1 1.2, Chapter 3, 3.1 & 3.2, Chapter 5, 5.1 to 5.3 and Chapter 6, 6.1 & 6.3.

8 Hours

UNIT II

Lossless compression algorithms: Introduction to lossless compression, Basic information theory, Run-length coding, Variable- length coding, Dictionary-Based Coding, Arithmetic coding, Lossless Image compression.
Text: Chapter 7, 7.1 to 7.7.

8 Hours

UNIT III

Lossy compression algorithms: Introduction to lossy compression, Distortion measures, Quantization, Transform coding, Wavelet-Based coding, Wavelet Packets, Embedded Zero tree of Wavelet Coefficients, SPIHT.
Text: Chapter 8, 8.1 to 8.9.

8 Hours

UNIT IV

Image Compression Standards and MPEG Video Coding I: The JPEG standard, The JPEG2000 standard, The JPEG-LS standard, Bi-level Image compression standard, H.261, Overview of MPEG-1. Text: Chapter 9, 9.1 to 9.4, Chapter 10, 10.1 to 10.4 and Chapter 11, 11.1 to 11.2
8 Hours

UNIT V
MPEG Video Coding II: MPEG-2, Overview MPEG-4, Object Based Visual Coding in MPEG-4, Synthetic Object coding in MPEG-4, MPEG-7. Text: Chapter 11, 11.3 Chapter 12, 12.1 to 12.3 & 12.6.
8 Hours

TEXT BOOKS		
1	Ze-Nian Li and Mark S. Drew	Fundamentals of Multimedia: First Edition, Pearson Education International, 2004.
2	Ze-Nian Li, Mark S. Drew and Jiangchuan Liu	Fundamentals of Multimedia: Second edition, Springer, 2014.

REFERENCE BOOKS		
1	Khalid Sayood	Introduction to Data Compression: Third Edition, Morgan Kaufmann Publishers.

Course Outcomes:	
Upon completion of this course the student will be able to:	
CO 1:	**Analyze** fundamental concepts of audio and video signals.
CO 2:	**Apply** lossless compression algorithms for text and image data.
CO 3:	**Apply** lossy compression algorithms for 1D and 2D signals.
CO 4:	**Analyze** image compression standards.
CO 5:	**Analyze** video compression standards like MPEG 1, 2, 4 and 7.

Object Oriented Programming with Java
(Engineering Science Course III-3)

Contact Hours/ Week:	: 3	Credits:	3
Total Lecture Hours:	: 40	CIE Marks:	50
Sub. Code:	: S3ISES03	SEE Marks:	50

Course objectives:
This course will enable students to:

1.	Introduction to object oriented programming.
2.	Formal introduction to Java programming language.
3.	Overall development of problem solving and critical analysis.

UNIT I

An Overview of Java: Object-Oriented Programming, A First Simple Program, A Second Short Program, Two Control Statements, Using Blocks of Code, Lexical Issues, The Java Class Libraries, Data Types, Variables, and Arrays: Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays, A Few Words About Strings.

8 Hours

UNIT II

Operators: Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses, Control Statements: Java‟s Selection Statements, Iteration Statements, Jump Statements.

8 Hours

UNIT III

Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection, The finalize( ) Method, A Stack Class, A Closer Look at Methods and Classes: Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Arrays Revisited, Inheritance: Inheritance, Using super, Creating a Multilevel Hierarchy, When Constructors are Called, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, The Object Class.

8 Hours

UNIT IV

Packages and Interfaces: Packages, Access Protection, Importing Packages, Interfaces, Exception Handling: Exception-Handling Fundamentals, Exception

Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java‟s Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Using Exceptions.
8 Hours

UNIT V
I/O programming: Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files. **Event handling and Multithreading:** Thread life cycle and methods, Runnable interface, Thread synchronization, Exception handling with try-catch-finally, Collections in java.
8 Hours

TEXT BOOKS	
1	Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2017.

REFERENCE BOOKS	
1	Introduction to Java Programming (Comprehensive Version), Daniel Liang, Seventh Edition, Pearson, 2008.
2	Cay S Horstmann, "Core Java - Vol. 1 Fundamentals", Pearson Education, 10th Edition, 2016.
3	Raoul-Gabriel Urma, Mario Fusco, Alan Mycroft, "Java 8 in Action", Dreamtech Press/Manning Press, 1st Edition, 2014.

Course Outcomes: Upon completion of this course the student will be able to:	
CO1.	Analyse and apply the basic principles of object-oriented programming.
CO2.	Show competency in using Java programming language to develop small to medium-sized application programs.
CO3.	Demonstrate the packages and interfaces of Java.
CO4.	Apply I/O programming to solve real world problems.
CO5.	Demonstrate the basic usage of multi-threaded programming.

Game Programming-2D
(Engineering Science Course III - 4)

Contact Hours/ Week:	3	Credits:	3
Total Lecture Hours:	40	CIE Marks:	50
Sub. Code:	S3ISES04	SEE Marks:	50

Course objectives:
This course will enable students to:

1.	This course introduces the fundamental concept of game programming.
2.	Discusses how programmers organize the objects in the games.
3.	It covers basics of vectors and physics involved in the gaming.
4.	It also covers how machines applies intelligent in the gaming.
5.	Discusses the graphics tools required to draw different sense of game.

UNIT I

Chapter 1: Game Programming Overview

Setting Up a Development Environment: Microsoft Windows, Apple macOS, Beyond the C++ Standard Library, The Game Loop and Game Class: Anatomy of a Frame, Implementing a Skeleton Game Class, Main Function, Basic Input Processing, Basic 2D Graphics: The Color Buffer, Double Buffering, Implementing Basic 2D Graphics, Drawing Walls, a Ball, and a Paddle, Updating the Game: Real Time and Game Time, Logic as a Function of Delta Time, Updating the Paddle's Position, Updating the Ball's Position, Game Project.

8 Hours

UNIT II

Chapter 2: Game Objects and 2D Graphics

Game Objects: Types of Game Objects, Game Object Models, Integrating Game Objects into the Game Loop, Sprites: Loading Image Files, Drawing Sprites, Animating Sprites, Scrolling Backgrounds, Game Project.

8 Hours

UNIT III

Chapter 3: Vectors and Basic Physics

Vectors: Getting a Vector between Two Points: Subtraction, Scaling a Vector: Scalar Multiplication, Combining Two Vectors: Addition, Determining a Distance: Length, Determining Directions: Unit Vectors and Normalization, Converting from an Angle to a Forward Vector, Converting a Forward Vector to an Angle: Arctangent, Determining the Angle between Two Vectors: Dot

Product, Calculating a Normal: Cross Product.
Basic Movement: Creating a Basic MoveComponent Class, Creating an InputComponent Class.

8 Hours

UNIT IV

Chapter 3: Vectors and Basic Physics Cont..

Newtonian Physics: Linear Mechanics Overview, Computing Positions with Euler Integration, Issues with Variable Time Steps.
Basic Collision Detection: Circle-Versus-Circle Intersection, Creating a CircleComponent Subclass, Game Project.

Chapter 4: Artificial Intelligence

Game Trees: Minimax, Handling Incomplete Game Trees, Alpha-Beta Pruning, Game Project.

8 Hours

UNIT V

Chapter 5: OpenGL

Initializing OpenGL: Setting Up the OpenGL Window, The OpenGL Context and Initializing GLEW, Rendering a Frame.
Triangle Basics: Why Polygons?, Normalized Device Coordinates, Vertex and Index Buffers.
Shaders: Vertex Shaders, Fragment Shaders, Writing Basic Shaders, Loading Shaders, Drawing Triangles.
Transformation Basics: Object Space, World Space, Transforming to World Space.
Matrices and Transformations: Matrix Multiplication, Transforming a Point by Using a Matrix, Transforming to World Space, Revisited, Adding World Transforms to Actor, Transforming from World Space to Clip Space, Updating Shaders to Use Transform Matrices.
Texture Mapping: Loading the Texture, Updating the Vertex Format, Updating the Shaders, Alpha Blending.
Game Project.

8 Hours

TEXT BOOKS

1	Sanjay Madhav	Game Programming in C++: Creating 3D Games, Addison-Wesley Professional, 2018.

REFERENCE BOOKS

1	Mike “MrMike” McShaffry and David “Rez” Graham	Game Coding Complete, Fourth Edition, Course Technology PTR: Cengage Learning, 2013.

Course Outcomes:	
Upon completion of this course the student will be able to:	
CO 1.	**Comprehend** the basic concepts of game programming.
CO 2.	**Apply** 2D graphics and objects involved in the game.
CO 3.	**Comprehend** vector operations and basics of physics required to implement games.
CO 4.	**Apply** artificial intelligent algorithms to solve game problems.
CO 5.	**Develop** 2D images or scenes required for the game using OpenGL.

Social Connect & Responsibilities

Contact Hours/ Week:	0:0:2:0	Credits:	1
Total Lecture Hours:	26	CIE Marks:	50
Sub. Code:	SHS01	SEE Marks:	50

Course objectives:
This course will enable students to:

1.	Enable the student to do a deep drive into societal challenges being addressed by NGO(s), social enterprises & The government and build solutions to alleviate these complex social problems through immersion, design & technology.
2.	Provide a formal platform for students to communicate and connect with their surroundings.
3.	Enable to create of a responsible connection with society.

Contents: The course is mainly activity-based that will offer a set of activities for the student that enables them to connect with fellow human beings, nature, society, and the world at large. The course will engage students in interactive sessions, reading groups and semester-long activities conducted by faculty mentors. In the following a set of activities planned for the course have been listed:

UNIT I

Plantation and adoption of a tree: Plantation of a tree by Miyawaki Method that will be adopted by entire semester by a group of students. They will also make an excerpt either as a documentary or a photoblog describing the plant's origin, its usage in daily life, and its appearance in folklore and literature.

6Hours

UNIT II

Heritage walk and crafts corner: Heritage tour, knowing the history and culture of the city, connecting to people around through their history, knowing the city and its craftsman, photoblog and documentary on evolution and practice of various craft forms.

6Hours

UNIT III

Organic farming: Definition of organic farming, Organically grown crops in India, Differentiate between conventional farming and organic farming, Necessity of organic farming, Key characteristics of organic farming, Four principles of organic farming(principle of Health, principle of ecology, principle of fairness and principle of care),Types of organic farming: 1) Pure organic farming, 2) Integrated farming (Integrated nutrient management and Integrated pest management), objectives of organic farming, benefits of organic farming, Basic steps in organic farming and limitations of organic

farming.
4 Hours

UNIT IV
Water Conservation: Global Water Scarcity - Global water crisis and its implications; Rainwater Harvesting - Concept and benefits of rainwater harvesting; Water Audit – An approach to water conservation; Efficient Water Use - Optimizing water consumption in daily life.
6 Hours

UNIT V
Food Walk City's culinary practices, food lore, and indigenous materials of the region used in cooking.
4 Hours

Course Outcomes:	
Upon completion of this course the student will be able to:	
CO 1.	Understand social responsibility
CO 2.	Practice sustainability and creativity
CO 3.	Showcase planning and organizational skills

Introduction to Blender Tool: Animation and 3D Creation
(Ability Enhancement Course III-1)

Contact Hours/ Week:	2P	Credits:	1
Total Lecture Hours:	26	CIE Marks:	50
Sub. Code:	S3ISA01	SEE Marks:	50

Course objectives:
This course will enable students to:

1.	Introduce the concept of Blinder Software tool.
2.	Creating an animation required for the applications.
3.	It also covers the creation of 3D views.

Perform the following using Blender tool:

1. Set up a screen with four viewports using a top, front, side and camera or perspective views and perform the following operations:
   i. Zooming
   ii. Changing window types
   iii. Centering the view on a certain object
   iv. Switching views (top, front, side, camera, free-rotate)
   v. Opening and closing the Tool Shelf and
   vi. Transform Panel

2. Create a sculpture using at least 1 of every type of mesh found in the Add-Mesh menu (do not use grid or circle). Remember to make sure you are in Object Mode before creating a new mesh. Use a plane for the ground and scale it large. Divide your 3D window into two so you can have one working view and one camera view. Use the RMB to select objects on the screen.

3. Create an object and perform the joining/separating meshes and boolean operations.

4. Develop a nice landscape scene using Landscape and Lighthouse of the Blender.

5. Create a 3D cloud for the background.

6. Apply the lighting and cameras concept that bounces off other objects, like real life on an object/s.

7. Create your own MPEG movie file.

TEXT BOOKS

1	James Chronister	Blender Basics Classroom Tutorial, 4th Edition, 2011.

REFERENCE BOOKS		
1	John M Blain	An Introduction to Blender 3D - A Book For Beginners, 2011
2.	Blender 3.2 Reference Manual, https://docs.blender.org/manual/en/latest/	

Course Outcomes:

Upon completion of this course the student will be able to:

CO1:	**Apply** the fundamental concepts of Blender tool.
CO2:	**Apply** the concept of creating and editing objects.
CO3:	**Design** an animation and add a 3D text for the applications.
CO4:	**Design** the objects using Texture, Rendering, and Lighting source for an image.

Documentation: MS Office and Latex
(Ability Enhancement Course III-2)

Contact Hours/ Week:	: 2P	Credits:	1
Total Practical Hours:	:26	CIE Marks:	50
Sub. Code:	: S3ISA02	SEE Marks:	50

Course objectives: This course will enable students to:	
1.	To create a professional quality MS Word document.
2.	To understand and use Cell computation functions and Formulas using Excel and create visible charts and Tables.
3.	To Understand usage of Latex for documentation.
4.	To create an IEEE Format Journal documentation and News Letter.

UNIT I

MS WORD: Text Basics, Text Formatting and saving file, Working with Objects, Header & Footers, working with bullets and numbered lists, Tables, Styles and Content.

6 Hours

UNIT II

MS WORD: Merging Documents, Sharing and Maintaining Document, Proofing the document, Printing.

MS EXCEL: Introduction to Excel, Formatting excel work book, Perform Calculations with Functions, Sort and Filter Data with Excel, Create Effective Charts to Present Data Visually.

6 Hours

UNIT III

MS EXCEL: Analyze Data Using Pivot Tables and Pivot Charts, Protecting and Sharing the work book, Use Macros to Automate Tasks.

MS POWERPOINT: Setting Up PowerPoint Environment, creating slides and applying themes, working with bullets and numbering, Working with Objects.

5 Hours

UNIT IV

MS POWERPOINT: Hyperlinks and Action Buttons, Using SmartArt and Tables, Animation and Slide Transition, Using slide Master, Slide show option.

4 Hours

UNIT V

Introduction to Latex: Installation of Latex and supporting tools, Formatting text matter, Creating tables, figures and graphs. Handling mathematical equations and symbols, preparation of Journal article, Preparation of report/thesis.

5 Hours

TEXT BOOKS		
1	Kumar Bittu	Mastering MS Office, V and S Publishers-2017 ISBN: 9789350578780, 9789350578780.
2	Dr. Sachin Vashisth and Hari Kishan Bhardwaj	Introduction LATEX and HTML, SHIVALIK PRAKASHAN, DELHI-2017. ISBN-10: 9788193451977, ISBN-13 : 978-8193451977.

REFERENCE BOOKS		
1	Lisa A. Bucki, John Walkenbach, Faithe Wempen	Microsoft Office 2013 Bible, Wiley Publishers, 2013. ISBN-10:8126543620 and ISBN-13:978-8126543625 .
2	Firuza Karmali Aibara	A Short Introduction to Latex: A Book for Beginners, Atlantic Publishers and Distributors. ISBN-10:1543162649 and ISBN-13:978-1543162646 .

Course Outcomes:	
Upon completion of this course the student will be able to:	
CO1:	Comprehend the basics of the Documentation using MS Word.
CO2:	Create excel sheets, perform computations, represent the results in graphical forms.
CO3:	Create power point presentations with animation, narration, and images.
CO4:	Develop and document IEEE journal template and newsletters using Latex.

UNIX and SHELL Programming
(Ability Enhancement Course III -3)

Contact Hours/Week	:	2P	Credits	:	1.0
Total Lecture Hours	:	--	CIE Marks	:	50
Total Practical Hours	:	26	SEE Marks	:	50
Course Code	:	S3ISA03			

Course objectives:
This course will enable students to:

1. Describe the architecture of Unix Operating System and demonstrate the various UNIX commands along with their options and arguments.
2. Apply suitable commands and filters for file processing.
3. Solve Text processing problems using Regular Expressions tools like Grep.
4. Develop shell scripts to automate tasks on a computer.

List of Problems

1. Introduction to Unix Operating System and demonstrate the basic general shell commands such as date, cal, who, uname, echo, bc, man, history, exit.
    - Display the time in GMT(Greenwich Mean Time)/UTC(Coordinated Universal Time)time zone.
    - Display the calendar of the complete current year with the current date highlighted.
    - Display list of users logged in to system.
    - Display all system information such as the kernel name, hostname, kernel version, operating system.
    - Display a string on the terminal.

    4 P Hours
2. Illustrate the basic directory commands like pwd, mkdir, cd, rmdir.
    - Create directories with name dir1, dir2, dir3 in root directory
    - Create directories d11, d12 under dir1 directory
    - Create directories d21, d22 under dir2 directory
    - Create files f11, f12 under dir1 directory
    - Create files f31, f32, file33 under dir3 directory
    - Navigate to directory dir3 and create folder d31 and files f31, f32
    - Demonstrate navigation from one directory to other using absolute path and relative path

    2 P Hours
3. Understand the basic file commands like cat, cp, mv, rm, ls, wc, cmp, chmod.
    - To view multiple files
    - Find out number of lines, word count, byte and characters count in a file.
    - Read and write file permission for Owner, and Read-only for the group and other.
    - Change the group ownership of a file or directory.

- Compare between two files and report the location of the first mismatch.

2 P Hours

4. Query a data file using filter commands in UNIX such as head, tail, cut, tr, nl, pr, join.
   - To view the first 20 lines of a file named "example.txt".
   - To view the last 26 lines of a file named "example1.txt".
   - Compare the ends of multiple files.
   - To print the version number of the command.
   - Combine the contents of these two files.
   - To convert characters from lower case to upper case.

4 P Hours

5. Search for a regular expression in a file using grep, sed command.
   - Find the number of lines that matches the given string.
   - Display the lines that are not matched with the specified search string.
   - Match the lines which start with the given string.
   - Match the lines which end with the given string.
   - Replaces the word “unix” with “linux” in the file.
   - Replace the first, second occurrence of a pattern in a line.

4 P Hours

6. Learn various features and controls of VI editor like edit commands, navigation commands, the ex-mode.
   - Opening an existing file with `file_name` = jayesh.
   - Copy lines or words from one place and paste them in another place.
   - Search some text for the string “Seeksforseeks”.
   - Delete the lines from 1 to 5.
   - Delete the lines from present cursor position to 3 lines.

2 P Hours

7. Understand shell programming and the conditional statements in it.
   - Conditional statement to check whether two numbers are equal or not equal.
   - Conditional statement to check whether given number is even or odd.

2 P Hours

8. Application of multi way branching in shell programming.
   - Given the car name, its headquarters address is to be given.
   - Design a simple calculator to perform basic arithmetic operations.

2 P Hours

9. Demonstration of numeric and string comparison in shell programming.
   - Compare the given numbers are equal or not.
   - Compare the strings are equal or not.

2 P Hours

10. Illustrate the different testing and loops in shell programming.
    - Test the validity of a command.
    - Check the type of file and the permissions related to a file.
    - Read a file with a for loop.
    - Read only a particular part of a file while loop.

2 P Hours

TEXT BOOKS		
1.	Sumitabha Das.	**Unix- concepts and applications**, Edition 4. Tata McGraw-Hill. 2017.

Course Outcomes:

Upon completion of this course the student will be able to:

CO1: Analyze the role of various components in the architecture of Unix Operating System and use various UNIX commands to interact with the operating system.

CO2: Design solutions for Text processing problems using Regular Expression tools like grep and sed.

CO3: Create files using vi editor and perform different editing operations.

CO4: Develop shell scripts to automate tasks on a computer.

Department of Information Science and Engg.

NEP 2022 BATCH

4th SEM SCHEME and SYLLABUS

A.Y: 2023-24

SCHEME OF TEACHING AND EXAMINATION (2022 Scheme)
IV Semester

Sl. No.	Course and Course Code		Course Title	Teaching / Paper setting Dept.	Teaching hrs./week				Examination				Credits
					Lecture	Tutorial	Practical/ Drawing	Self-Study Component	Duration in hrs.	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1.	PCC	S4IS01	Operating System	IS	3	0	0		3	50	50	100	3
2.	IPCC	S4ISI01	Design and Analysis of Algorithms (Integrated)	IS	3	0	2		3	50	50	100	4
3.	IPCC	S4ISI02	ARM Processor and Microcontroller (Integrated)	IS	3	0	2		3	50	50	100	4
4.	PCCL	S4ISL01	Data Visualization using R Programming Laboratory	IS	0	0	2		3	50	50	100	1
5.	ESC	S4ISSCXX	ESC/ETC/PLC	IS	3	0	0		3	50	50	100	3
6.	BSC	S4CCA01	Biology for Engineers (Board: BT)	BT	3	0	0		3	50	50	100	3
7.	UHV	SHS02	Universal Human Values Course (Board: IEM)	IEM	1	0	0		1½	50	50	100	1
8.	AEC/ SEC	S4ISAXX	Ability Enhancement Course/ Skill Enhancement Course – IV	IS	If offered as Theory Course: 1	0	0		1½	50	50	100	1
					If offered as Integrated Course: 0	0	2		1½				
9.	NCMC	SMC01	National Service Scheme (NSS)	NSS CO	0	0	2			100	-	100	0
		SMC02	Physical Education (PE) (Sports and Athletics)	PED									
		SMC03	Yoga	PED									
			Total							**500**	**400**	**900**	**20**
		AAP	AICTE Activity Points (Applicable for both Regular and Lateral Entry students)	40 hours community service to be documented and produced for the examination									

Note: PCC: Professional Core Course, IPCC: Integrated Professional Core Course, PCCL: Professional Core Course laboratory, UHV: Universal Human Value Course, NCMC: Non Credit Mandatory Course, AEC: Ability Enhancement Course, SEC: Skill Enhancement Course, ESC: Engineering Science Course, ETC: Emerging Technology Course, PLC: Programming Language Course
L: Lecture, T: Tutorial, P: Practical S= SDA: Skill Development Activity, CIE: Continuous Internal Evaluation, SEE: Semester End Evaluation.

Engineering Science Course (ESC/ETC/PLC) (Offered by the Department)

S4ISES01	Discrete Mathematics Structures	S4ISES02	Statistical Computation
S4ISES03	Theory of Computation	S4ISES04	Game Programming-3D

Ability Enhancement Course – IV (Offered by the Department)

S4ISA01	Swift Programming	S4ISA02	Mobile Application Development
S4ISA03	Java Application Development for Healthcare and Agriculture	S4ISA04	Natural Language Processing

Professional Core Course (IPCC): Refers to Professional Core Course Theory Integrated with practical of the same course. Credit for IPCC can be 04 and its Teaching–Learning hours (L : T : P) can be considered as (3 : 0 : 2) or (2 : 2 : 2). The theory part of the IPCC shall be evaluated both by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper. For more details, the regulation governing the Degree of Bachelor of Engineering (B.E.) 2022-23 may please be referred.

National Service Scheme /Physical Education/Yoga: All students have to register for any one of the courses namely National Service Scheme (NSS), Physical Education (PE)(Sports and Athletics), and Yoga(YOG) with the concerned coordinator of the course during the first Week of III semesters. Activities shall be carried out between III semester to the VI semester (for 4 semesters). Successful completion of the registered course and requisite CIE score is mandatory for the award of the Degree. The events shall be appropriately scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities. These courses shall not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the course is mandatory for the award of Degree.

Operating Systems

Contact Hours/Week	:	3	Credits	:	3.0
Total Lecture Hours	:	40	CIE Marks	:	50
Total Tutorial Hours	:	00	SEE Marks	:	50
Course Code	:	S4IS01			

Course objectives:
This course will enable students to:

1.	Define fundamental OS abstractions such as processes, threads, files etc.
2.	Visualize the intricate relationship between an operating system and its underlying hardware.
3.	Explain scheduling algorithms, deadlock detection algorithms and memory management strategies.
4.	Apply the principles of concurrency and synchronization, to write concurrent programs/ software.

UNIT I

INTRODUCTION: What operating systems do - User view, System view, Defining operating systems, Operating System Structure, Operating System Operations – Dual mode and multi-mode operation, Timer, Process Management; Memory Management; Storage Management; Protection and Security. [Textbook 1:chapteres 1.1, 1.4 to 1.9]

SYSTEM STRUCTURES: Operating System Services; System calls; Types of system calls; System programs; Operating System Structure –Simple structure, Layered approach, Micro kernels, Modules [Textbook 1: Chapters 2.1, 2.3 to 2.5, 2.7.4]

Self study : Hybrid Systems – Mac OS X, iOS, Android.[Textbook 1: Chapter 2.7.5]

8 Hours

UNIT II

PROCESS: Process concept, Process state, Process control block, Process scheduling, Scheduling queues, Schedulers, Context switch, Operations on processes – Process creation and termination, Inter-process communication, Shared memory and message passing systems. [Textbook 1: Chapters 3.1 to 3.4]

PROCESS MANAGEMENT: Basic concepts, CPU scheduler, Preemptive and non-preemptive scheduling, Scheduling criteria, Scheduling algorithms – FCFS, SJF, Priority and Round robin scheduling. [Textbook 1: Chapters 6.1 to 6.3.4]

Self Study : Multi-level and multilevel feedback queue scheduling.[Textbook 1: Chapter 6.3.5]

8 Hours

UNIT III

THREADS: Overview, Benefits, Multi core Programming, Types of parallelism, Multi threading models. [Textbook 1: Chapters 4.1-4.3]

PROCESS SYNCHRONIZATION: Background, The Critical section problem,

Peterson's solution, Synchronization hardware, Mutex locks, Semaphores, Classic problems of synchronization, Bounded buffer problem. [Textbook 1: Chapters 5.1 to 5.7.1]

Self-Study: Readers writer's problem, Dining philosopher's problem. [Textbook 1:Chapters 5.7.2 , 5.7.3]

8 Hours

UNIT IV

DEADLOCKS: System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection and recovery from deadlock. [Textbook 1:7.1-7.7]

MEMORY MANAGEMENT: Background, Basic hardware, Address binding, Logical and physical address, swapping, Dynamic loading and linking. [Textbook 1: Chapters 8.1, 8.2]

8 Hours

UNIT V

MEMORY MANAGEMENT: Contiguous memory allocation, Segmentation, Paging. [Textbook 1: 8.3, 8.4, 8.5]

VIRTUAL MEMORY MANAGEMENT: Basic concepts, Demand paging, Copy-on-write, Page replacement – FIFO, LRU, Optimal. [Textbook 1:9.1-9.4]

Self-Study: Structure of page table, Hierarchical paging, Hashed paging, Inverted paging. [Textbook 1: Chapter 8.6]

8 Hours

NOTE: Self-study topics are assessed only in CIE-Quizzes/ Assignment

TEXT BOOKS

1	Abraham Silberschatz, Peter Baer Galvin, Greg Gagne	Operating System Concepts Wiley-India,9th Edition, 2013.

REFERENCE BOOKS

1	D.M Dhamdhere	Operating System - A Concept Based Approach, Tata McGraw- Hill, 2nd Edition, 2002.
2.	P.C.P. Bhatt	Operating Systems, PHI 4th Edition, 2013.

Course Outcomes:

Upon completion of this course the student will be able to:

CO1: **Identify** the services, functions and structure of different operating systems.

CO2: **Apply and analyze** appropriate scheduling algorithm for process selection and execution.

CO3: **Identify and analyze** the techniques used to solve process synchronization issues.

CO4: **Apply and analyze** various deadlock prevention, avoidance, detection and recovery mechanisms to solve real world problems.

CO5: **Analyze** the performance of various memory management techniques and page replacement algorithms.

Design and Analysis of Algorithms
(Integrated Course)

Contact Hours/Week	:	3L + 2P	Credits	:	4.0
Total Lecture Hours	:	40	CIE Marks	:	50
Total Practical Hours	:	26	SEE Marks	:	50
Course Code	:	S4ISI01			

Course objectives:

This course will enable students to:

1. Understand fundamentals of design and analysis of algorithms.
2. Analyze the running time of the basic algorithms and prove their correctness.
3. Compare the running time of different sorting and searching algorithms.
4. Apply transform and conquer, dynamic programming and greedy design techniques to solve problems.
5. Understand the fundamental principles of space-and-time tradeoffs and backtracking techniques.

UNIT I

INTRODUCTION: Notion of algorithm, Fundamentals of Algorithmic Problem Solving, Fundamentals of the Analysis of Algorithm Efficiency: Analysis frame work, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non-recursive and Recursive Algorithms. [Chapters: 1.1, 1.2, 2.1-2.4]

08 Hours.

UNIT II

BRUTE FORCE: Selection Sort and Bubble Sort, Sequential Search and Brute-Force String Matching. Divide and Conquer: Mergesort, Quicksort, Binary Search. [Chapters: 3.1, 3.2, 4.1- 4.3]

08 Hours.

UNIT III

DECREASE AND CONQUER: Insertion Sort, Depth First Search, Breadth First Search, Topological Sorting, Algorithms for Generating Combinatorial Objects.

TRANSFORM AND CONQUER: Presorting, Balanced Search Trees: AVL Tree, Heaps and Heapsort. [Chapters: 5.1 – 5.4, 6.1, 6.3 (only AVL Trees),6.4]

08 Hours.

UNIT IV

DYNAMIC PROGRAMMING: Computing a Binomial Coefficient, Warshall's and Floyd's Algorithms, The Knapsack Problem.

GREEDY TECHNIQUE: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm. [Chapters:8.1, 8.2, 8.4, 9.1-9.3]

08 Hours.

UNIT V

SPACE AND TIME TRADEOFFS: Sorting by Counting, Input Enhancement in String Matching : Horspool's Algorithm.
LIMITATIONS OF ALGORITHM POWER: P, NP and NP-Complete Problems. Coping with the Limitations of Algorithm Power: Backtracking: N-Queens, Hamiltonian Circuit Problem, Subset-Sum Problem. [Chapters: 7.1, 7.2, 10.3, 11.1]

08 Hours.

Self-Study Component:

The following topics must be studied by the students:

1. Algorithms for Generating Combinatorial Objects.
2. Limitations of Algorithm Power: P, NP and NP-Complete Problems.

NOTE: Self-study topics are assessed only in CIE-Quizzes/ Assignments

TEXT BOOKS:

1.	Anany Levitin.	Introduction to The Design & Analysis of Algorithms. Edition 2. Pearson Education. 2011.

REFERENCE BOOKS:

1.	Ellis Horowitz, SatrajSahni and Rajasekharan.	Fundamentals of Computer Algorithms, University Press Pvt. Ltd, Edition 2, 2009.

List of Problems for laboratory:

1. Programs on Brute-Force technique.
   a) Develop a recursive program to implement Linear Search.
   b) Develop a program to find the maximum and minimum in a given list of n elements.

2. Programs on Divide and Conquer technique.
   a) Develop a program to implement Merge Sort.
   b) Develop a program to implement Quick Sort.

3. Programs on Decrease and Conquer technique.
   a) Develop a program to demonstrate Topological ordering of vertices in a given digraph.
   b) Develop a program to implement Insertion sort algorithm to sort a given set of elements.
   c) Develop a program to Print all nodes reachable from a given node in a digraph using Breadth First Search and Depth First Search.

4. Program on Transform and Conquer technique.
   a) Develop a program to implement Heap sort.

5. Programs on Dynamic Programming.
   a) Develop a program to implement 0/1 Knapsack problem.
   b) Develop a program to Compute the transitive closure of a given graph using Warshall's algorithm.
   c) Develop a program to implement All pair shortest path problem using Floyd's algorithm.
   d) Find the Binomial Coefficient using dynamic programming.

6. Programs on Greedy Technique.
   a) Develop a program to implement Prim's algorithm to find minimum cost spanning tree of a given weighted graph.
   b) Develop a program to implement Kruskal's algorithm to find minimum cost spanning tree of a given weighted graph.

7. Programs on Space and Time trade-off.
   a) Develop a program to demonstrate string matching using Horspool's algorithm.
   b) Develop a program to perform Comparison counting sorting on a list of elements.

8. Programs on Backtracking.
   a) Write a program to find a subset of a given set.
   S = {s1, s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d.
   For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions {1,2,6} and {1,8}.

   b) Write a program to solve N-Queen's problem.

Course Outcomes:

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

CO1: **Identify** the fundamental principles of algorithm analysis and design and apply them in specific instances.

CO2: **Apply** design techniques such as brute force and divide-and-conquer to solve a given problem.

CO3: **Apply** design techniques such as decrease and conquer, transform and conquer to provide solutions for specific problems.

CO4: **Apply** dynamic programming and greedy techniques to find optimal solutions to given problems.

CO5: **Identify** the fundamental principles of space-and-time trade-offs and apply the design techniques such as backtracking to solve a given problem.

ARM Processor and Microcontroller (Integrated Course)

Contact Hours/ Week:	3T + 2P	Credits:	4.0
Total Lecture Hours:	40 + 26	CIE Marks:	50
Sub. Code:	S4ISI02	SEE Marks:	50

Course objectives:
This course will enable students to:

1.	Introduce the concept of architecture and programming of advanced embedded microcontrollers.
2.	ARM family of microcontrollers that are widely used in design of real time sophisticated embedded systems like tablets, hand held devices, automation and industrial control systems.
3.	It also covers writing Embedded C programming of LPC2148 for GPIO, ADC, DAC, UART, LCD, Timers and etc.
4.	It also explains the concepts of embedded system and its components.

UNIT I

CHAPTER 1-EMBEDDED SYSTEM COMPONENTS: Embedded v/s General computing system, Classification of Embedded systems, Major applications and purpose of Embedded systems, Core of an Embedded System including all types of Processors, Controller and Memory.

CHAPTER 2-ARM EMBEDDED SYSTEMS: The RISC Design Philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software.

8 Hours

UNIT II

CHAPTER 3-ARM PROCESSOR FUNDAMENTALS: Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and Vector Table, Core Extensions, Architecture Revisions, ARM Processor Families, LPC2148 Microcontroller Architecture, LPC2148 Microcontroller Architecture, Memory Mapping, Register Description.

8 Hours

UNIT III

CHAPTER 4-INTRODUCTION TO THE ARM INSTRUCTIONS SET: Data Processing Instructions and examples, Branch Instructions and examples, Load-Store Instructions and examples, Software Interrupt Instructions and examples, Program Status Register Instruction and examples. Programs: Addition, Multiplication, division and Subtraction of 16, 32-bit data, Example Programs: Looping, conditional programs like sum of memory elements, Loading Constants, Conditional Execution, and Example Programs.

8 Hours

UNIT IV
CHAPTER 5: INTERFACING Sensors, Actuators, GPIO, LED interfacing and programming in C, 7 segment display interfacing and C program, stepper motor interfacing, Keyboard interfacing, Push button switch interfacing and programming in C.
8 Hours
UNIT V
INTERFACING Continued... Data Conversions (ADC, DAC), LCD interfacing and C program, Timers, Counters, Communication Protocols: UART, CAN Programs using C.
8 Hours

TEXT BOOKS	
1	Andrew N. Sloss, Dominic Symes, Chris Wright, ARM Systems Developer's Guide Designing and Optimizing System Software, Morgan Kaufmann Publishers, Elsevier Inc, 2004(Chapters 1, 2, 3)
2	Shibu K V, Introduction to Embedded Systems, Second Edition, Tata McGraw Hill Education Private Limited, 2017. (Chapters 1 and 2 selected topics)
3	LPC214x User Manual –http://www.keil.com/dd/docs/datashts/philips/ (LPC2148, GPIO, Registers, Embedded components selected)

REFERENCE BOOKS	
1	Steve Furber, ARM System on Chip Architecture, Second Edition, Pearson Education Limited, 2000.
2	William Hohl, Christopher Hinds, ARM Assembly Language Fundamentals and Techniques, Second Edition, CRC Press, 2015.
3	Gibson, ARM Assembly Language an Introduction, Second Edition, 2007.

List of Problems for laboratory:

1. Write an ALP to multiply two binary numbers.
2. Write an ALP to find the sum of first 10 integer numbers.
3. Write an ALP to find factorial of a number.
4. Write an ALP to add an array of numbers and store the result in internal RAM.
5. Write an ALP to find the square of a number (1 to 10) using look-up table.
6. Write an ALP to find the largest/smallest number in an array of 32 numbers.
7. Write an ALP to arrange a series of 32-bit numbers in ascending/descending order.

8. Write an ALP to count the number of ones and zeros in two consecutive memory locations.

Conduct the following experiments on LPC2148 evaluation board using evaluation version of Embedded C and Keil µvision tool/compiler.

9. Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.
10. Interface a DAC and generate the following waveforms:
    a. Triangular b. Square c. Sin wave
11. Interface a 3x8 keyboard and display the key pressed on an LCD.
12. Determine Digital output for a given Analog input using Internal ADC of ARM controller.
13. Demonstrate the use of a Logic controller interface to toggle an LED On/Off.
14. Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in between the digits.
15. Display "Hello World" message using Internal UART.

Course Outcomes:

Upon completion of this course the student will be able to:

CO 1.	**Describe** the fundamental concept of Embedded System Architecture.
CO 2.	**Describe** the ARM processor architecture and its family.
CO 3.	**Develop** assembly language programs to perform specific tasks using ARM instructions.
CO 4.	**Design** and **develop** embedded C programs to interface external hardware with LPC214x microcontroller.
CO 5.	**Design** and **develop** the solutions for a problem using embedded system and demonstrate.

Data Visualization Using R Programming Laboratory

Contact Hours/Week	:	2P	Credits	:	1.0
Total Lecture Hours	:	--	CIE Marks	:	50
Total Practical Hours	:	26	SEE Marks	:	50
Course Code	:	S4ISL01			

Course objectives:

This course will enable students to:

1. Understand different ways of representation of data.
2. Design and create data visualization plots.
3. Conduct exploratory data analysis using data visualization.

List of Problems for laboratory

I Data Visualization

1. Introduction to R Programming: Overview of R Programming. Introduction to data visualization, Downloading and installing, Help of Function, Viewing documentation, General issues in R, Package Management.

4 Hours

2. Data Inputting in R.

a. Data Types
b. Sub-setting
c. Writing data
d. Reading from csv files
e. Creating a vector and vector operation
f. Initializing data frame

4 Hours

3. Exploring the data before visualization:

a. Exploring Datasets Available in the R Base Package.
b. Introduction to ggplot2 Package .

2 Hours

4. Ggplot for plots and graphs: Creating bar chart, dot plot, Histograms.

2 Hours

5. Data analysis using Boxplots.

2 Hours

6. Scatter plots for 2D and 3D data set.

7. Heat map and Map visualization.

2 Hours

II. Application-based data visualization:

1. Plotting single continuous variable, Plotting single discrete variable.
2. Plotting two continuous variables, Plotting one continuous and one discrete variable.

4 Hours

3. Weather forecasting.
4. Social Networking.

2 Hours

5. Visualizing clusters using Dendograms.
6. Remote sensing data analysis.

4 Hours

REFERENCES:		
1.	Rahlf Thomas	Data Visualisation with R, Edition 2, 2015, Springer Nature Switzerland AG.
2.	Tony Fischetti, Brett Lantz, Jaynal Abedin, Hrishi V. Mittal, et.al.	R: Data Analysis and Visualization, Edition 1, Packt Publishing, 2016.
3.	https://onlinecourses.nptel.ac.in/noc22_cs72/preview	
4.	https://www.rstudio.com/blog/r-and-rstudio-the-interoperability-environment-for-data-analytics/	
5.	https://datacarpentry.org/R-ecology-lesson/04-visualization-ggplot2.html	

Course Outcomes:

Upon completion of this course the student will be able to:

CO1: **Design and generate** different types of plots for various types of data available.

CO2: **Apply** appropriate data visualization methods to analyze real-time application data.

CO3: **Analyze** and **Evaluate** the effectiveness of data visualization to various domains of data analytics.

Discrete Mathematics Structures
(Engineering Science Course-1)

Contact Hours/Week	:	3	Credits	:	3
Total Lecture Hours	:	40	CIE Marks	:	50
Total Tutorial Hours	:	-	SEE Marks	:	50
Course Code	:	S4ISES01			

Prerequisites: Set Theory.

Course objectives:

1. To identify the domain and range of a relation and its properties, use function notation and evaluate function.
2. To understand the mathematical application of symmetry to an object to obtain knowledge of its physical properties.
3. To identify and apply various properties of and relating to the integers including the well-ordering principle, prime, unique factorization, the division algorithm and greatest common divisor.
4. To compute multiplication inverse, expressing the whole in parts and test of primitive.
5. To study the theory of Boolean algebra and to representation of switching functions using Boolean expressions and their minimization technique.

UNIT-I

Relations and Function: Relations, Properties of Relations, Computer Recognition- Zero-One Matrices and Digraphs, Partial order relation -Poset and Hasse-Diagrams, Equivalence Relation and Partitions, Extremal elements of a Poset, Lattice.

8 Hours

UNIT-II

Groups: Binary Operations and Properties, Definition of a Group, Examples and Elementary properties, Abelian Groups, Homomorphism, Isomorphism and Cyclic Groups, Cosets and Lagrange's Thoerem, Normal subgroups.

8 Hours

UNIT- III

Number Theory – Divisibility Theory in Integers: Introduction, The division algorithm, greatest common divisor, Euclidean Algorithm, The Diophantine equation ax + by = c, Fundamental theorem of arithmetic, The Goldbach conjecture.

8 Hours

UNIT-IV

The Theory of congruences: Basic properties of congruences, Binary and decimal representation of integers, Chinese remainder theorem, Fermat's Theorem, Wilson Theorem, The Fermat-Kraitchik Factorization method.

8 Hours

UNIT-V

Boolean Algebra and Switching Functions: Introduction, Definition of Boolean algebra and Boolean function, Laws of Boolean functions and problems Switching functions: Disjunctive and conjunctive normal forms. Structure of Boolean Algebra.

8 Hours

TEXT BOOKS:

1. Ralph P. Grimaldi, "Discrete and Combinatorial Mathematics", 5th Edition, *Pearson Education*, 2012.
2. Bernard Kolman, Robert Busby and Sharon C. Ross, "Discrete Mathematical Structures", 6th edition, Pearson Education, 2012.
3. David M Burton, "Elementary Number Theory", 7th Edition, McGraw Hill Education, 2013.

REFERENCE BOOKS:

1. Kenneth H. Rosen, "Discrete Mathematical and its Applications", Tata-McGrawHill, 7th Edition-2011.
2. J.P.Tremblay and R. Manohar, "Discrete Mathematical Structures with Applications to computer science", Tata-McGraw Hill, 2010.
3. M. Ram Murthy and Jody Esmonde, "Problems in Algebraic number theory", Springer, 2006.
4. Erwin Kreyszig, "Advanced Engineering Mathematics" , 10th edition, Wiley Publications, 2015.

Course Outcomes:

Upon completion of this course the student will be able to:

CO1: Derive logical implications and equivalences using laws of logic, describe use quantifiers and prove given statement in different ways. (L1, L2).

CO2: Compute zero-one matrix, composition of relations and draw Hasse diagram (L3).

CO3: Explain the concept of groups, subgroup, Abelian group and derive Lagrange's theorem in groups (L2).

CO4: Determine gcd by different methods and represent gcd as a linear combination. Euclidean algorithm and its applications (L3).

CO5: Perform congruence arithmetic. Compute inverse mod p using different methods and know the existence of primitive roots (L3).

Statistical Computation
(Engineering Science Course-2)

Contact Hours/Week	:	3	Credits	:	3
Total Lecture Hours	:	40	CIE Marks	:	50
Total Tutorial Hours	:	-	SEE Marks	:	50
Course Code	:	S4ISE02			

Course objectives: This course will enable students to:	
1.	Define elements structure and variability.
2.	Visualize the sampling criteria and hypothesis using different algorithms.
3.	Explain regression and prediction techniques for computing.
4.	Apply the principles of computing for regression, prediction and classifications.

UNIT 1

EXPLORATORY ANALYSIS: Elements of Structured, Estimates of Location - Mean, Median, Mode, Outliers, Estimates of Variability- Standard Deviation, Z-Score, Frequency Table and Histograms, Correlation.

8 Hours

UNIT 2

DATA SAMPLING AND DISTRIBUTION: Normalization, Sampling Data- Simple Random sampling, Stratified, Cluster Sampling, Sampling Error/Bias. Bootstrapping, Central Limit Theorem, Confidence intervals, Normal distribution, Binomial distribution, Poisson distribution.

8 Hours

UNIT 3

HYPOTHESIS: A/B Testing, Hypothesis Tests- null, one-way, two-way, P-value, Type 1 & 2 errors, t-tests, multiple testing, degrees of freedom, ANOVA, Chi-Square Tests, Power and Sample Size.

8 Hours

UNIT 4

REGRESSION AND PREDICTION: Simple Linear Regression, Multiple Linear Regression, Confidence and Prediction Intervals, Categorical Variables, Multicollinearity, Polynomial Regression.

8 Hours

UNIT 5

CLASSIFICATION: Naive Bayes, Discriminant Analysis, Logistic Regression, Evaluating Classification Models, Strategies for Imbalanced Data.

8 Hours

TEXT BOOKS

1. Bruce, Peter, and Andrew Bruce. Practical statistics for data scientists: 50 essential concepts. " O'Reilly Media, Inc.", 2017.

REFERENCE BOOKS

1. Dodge, Yadolah, ed. Statistical data analysis and inference. Elsevier, 2014.
2. Ismay, Chester, and Albert Y. Kim. Statistical Inference via Data Science: A Modern Dive into R and the Tidyverse. CRC Press, 2019.

E BOOKS

1. https://leanpub.com/LittleInferenceBook

Course Outcomes: Upon completion of this course the student will be able to:	
CO 1.	**Perform** exploratory analysis on the datasets.
CO 2.	**Comprehend** the various distribution and sampling.
CO 3.	**Perform** Hypothesis Testing on datasets.
CO 4.	**Apply** statistical inference for Regression.
CO 5.	**Apply** statistical inference for Classification.

Theory of Computation
(Engineering Science Course-3)

Contact Hours/ Week:	3	Credits:	3.0
Total Lecture Hours:	40	CIE Marks:	50
Sub. Code:	S4ISES03	SEE Marks:	50

Course objectives:
This course will enable students to:

1.	Describe the concepts of automata theory and formal languages.
2.	Identify different formal language classes like regular and context free and their relationships.
3.	Design Regular expression, grammars and recognizers for different formal languages.
4.	Describe Turing machine, its variants and hierarchy of formal languages and automata.

UNIT I

Introduction to Finite Automata: The central concepts of Automata theory; Deterministic finite automata; Nondeterministic finite automata; An Application: Text search; Finite automata with Epsilon-transitions; Equivalence and Minimization of Automata- Testing equivalence of states, Testing equivalence of regular languages, Minimization of DFA's. (Text Book 1- 1.1.1, 1.5, 2.2, 2.3, 2.4, 2.5, 4.4.1-4.4.3) [Theorem: 2.11]

8 Hours

UNIT II

Regular expressions: Finite Automata and Regular Expressions- From DFA's to Regular Expressions, Converting DFA's to Regular expressions by eliminating states, Converting regular expressions to automata; Application of Regular Expressions, Properties of Regular languages- Proving languages not to be regular languages; Closure properties of regular languages- Boolean operations, reversal, homomorphisms. (Text Book 1- 3.1, 3.2.1, 3.2.2, 3.2.3, 3.3, 4.1, 4.2.1- 4.2.3) [Theorems: 3.4,3.7,4.1,4.4,4.5,4.8,4.10,4.11]

8 Hours

UNIT III

Context-Free Grammars and Languages:
Context –free grammars; Parse trees- Constructing Parse Trees, The yield of a parse tree; Applications of Context Free Grammars, Ambiguity in grammars and languages- Ambiguous grammars, Leftmost derivation as a way to express ambiguity, Inherent ambiguity. (Text Book 1- 5.1, 5.2.1, 5.2.2,5.3, 5.4.1, 5.4.3,5.4.4)
Simplification of CFG's and Normal forms (Text Book 2, chapter 6- 6.1,6.2).

[Theorems: 6.1,6.2,6.3,6.4,6.5,6.6]

8 Hours

UNIT IV

Pushdown Automata:

Definition of the Pushdown automata; The languages of a PDA; Equivalence of PDA's and CFG's; Deterministic Pushdown Automata- Definition of a DPDA. (Text Book 1- 6.1, 6.2, 6.3, 6.4.1) [Theorems: 6.9,6.11]

8 Hours

UNIT V

Introduction To Turing Machine:

The Turing Machine: Notation for the TM, Instantaneous Descriptions for the TM, Transition diagrams for the TM, The Language of a TM, TM and Halting; A hierarchy of Formal Languages and Automata- Definitions of Recursive and Recursively Enumerable Languages, Definition of Unrestricted Grammars, Definition of Context Sensitive Grammars and Languages, Chomsky hierarchy. (Text Book 1, Chapter 8- 8.2.2-8.2.6, Text Book 2, Chapter 11- 11.1, 11.2, 11.3, 11.4 only mentioned definitions of chapter 11).

8 Hours

TEXT BOOKS

1	John E.. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman	Introduction to Automata Theory, Languages and Computation, 3rd Edition, Pearson education, 2007.
2	Peter Linz	An Introduction to Formal Languages and Automata, 4th edition, Narosa publication.

REFERENCE BOOKS

1	John C Martin	Introduction to Languages and Automata Theory, 3rd Edition, Tata McGraw-Hill, 2007.

Course Outcomes:

Upon completion of this course the student will be able to:

CO 1.	**Apply** the basic mathematical properties to understand grammars, automata theory and formal languages.
CO 2.	**Apply** the automata theory to show the equivalence among different notations of regular and context free languages.
CO 3.	**Design** the regular expressions and context free grammars for a given language.
CO 4.	**Apply** the properties of pumping lemma, norm forms, context free grammars and regular grammar to prove the properties of a given languages.
CO 5.	**Design** the finite automata, pushdown automata and Turing machine for a given problem.

Game Programming-3D
(Engineering Science Course-4)

Contact Hours/ Week:	3	Credits:	3.0
Total Lecture Hours:	40	CIE Marks:	50
Sub. Code:	S4ISES04	SEE Marks:	50

Course objectives:
This course will enable students to:

1.	This course introduces the fundamental concept of 3 graphics.
2.	Discusses how programmers can choose the different 3D models.
3.	It covers basics of 3D Meshes and Lighting involved in 3D gaming.
4.	It also covers basic listener to include the audio in the gaming.
5.	Discusses the animation of graphics and updating.

UNIT I

3D Graphics: The Actor Transform in 3D, Transform Matrices for 3D, Euler Angles, Quaternions, New Actor Transform in Action.
Loading 3D Models: Choosing a Model Format, Updating the Vertex Attributes, Loading a gpmesh File.

8 Hours

UNIT II

Drawing 3D Meshes: Transforming to Clip Space, Revisited, Out with the Painter's Algorithm, in with ZBuffering, The BasicMesh Shader, The MeshComponent Class.

8 Hours

UNIT III

Lighting: Revisiting Vertex Attributes, Types of Lights, Phong Reflection Model, Implementing Lighting.

8 Hours

UNIT IV

3D Positional Audio: Setting Up a Basic Listener, Adding Positional Functionality to SoundEvent, Creating an AudioComponent to Associate Actors with Sound Events, The Listener in a Third-Person Game, The Doppler Effect.

8 Hours

UNIT V

Skeletal Animation: Foundations of Skeletal Animation, Skeletons and Poses, The Inverse Bind Pose Matrix, Animation Data, Skinning. **Implementing Skeletal Animation:** Drawing with Skinning Vertex Attributes, Loading a Skeleton, Loading the Animation Data, The Skinning Vertex Shader, Updating Animations.
8 Hours

TEXT BOOKS		
1	Sanjay Madhav	Game Programming in C++: Creating 3D Games, Addison-Wesley Professional, 2018.

REFERENCE BOOKS		
1	Mike "MrMike" McShaffry and David "Rez" Graham	Game Coding Complete, Fourth Edition, Course Technology PTR: Cengage Learning, 2013.

Course Outcomes: Upon completion of this course the student will be able to:	
CO 1.	**Comprehend** the basic concepts of 3D graphics and loading models.
CO 2.	**Apply** the audio effects for the games.
CO 3.	**Creating** the 3D mesh using inbuilt models and algorithms.
CO 4.	**Describe** the light models and implementing.
CO 5.	**Develop** 3D animation and updating the animation based on the feedback.

Biology for Engineers

Contact Hours/ Week:	3+0+0 (L+T+P)	Credits:	3.0
Total Lecture Hours:	40	CIE Marks:	50
Sub. Code:	S4CCA01	SEE Marks:	50

Course objectives: This course will enable students to:	
1.	To familiarize the students with the basic biological concepts and their engineering applications.
2.	To enable the students with an understanding of biodesign principles to create novel devices and structures.
3.	To provide the students an appreciation of how biological systems can be re-designed as substitute products for natural systems.
4.	To motivate the students to develop interdisciplinary vision of biological engineering.

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

- Explanation via real life problem, situation modelling, and deliberation of solutions, hands-on sessions, reflective and questioning /inquiry-based teaching.
- Instructions with interactions in classroom lectures (physical/hybrid).
- Use of ICT tools, including YouTube videos, related MOOCs, AR/VR/MR tools.
- Flipped classroom sessions (~10% of the classes).
- Industrial visits, Guests talks and competitions for learning beyond the syllabus.
- Students' participation through audio-video based content creation for the syllabus (as assignments).
- Use of gamification tools (in both physical/hybrid classes) for creative learning outcomes.
- Students' seminars (in solo or group) /oral presentations.

UNIT I
Introduction To Biology: The cell: the basic unit of life, Structure and functions of a cell. The Plant Cell and animal cell, Prokaryotic and Eukaryotic cell, Stem cells and their application. Biomolecules:

Properties and functions of Carbohydrates, Nucleic acids, proteins, lipids. Importance of special biomolecules; Enzymes (Classification (with one example each), Properties and functions), vitamins and hormones.

8 Hours

UNIT II

Biomolecules and Their Applications (Qualitative): Carbohydrates (cellulose-based water filters, PHA and PLA as bioplastics), Nucleic acids (DNA Vaccine for Rabies and RNA vaccines for Covid19, Forensics – DNA fingerprinting), Proteins (Proteins as food – whey protein and meat analogs, Plant based proteins), lipids (biodiesel, cleaning agents/ detergents), Enzymes (glucose-oxidase in biosensors, lignolytic enzyme in bio-bleaching).

8 Hours

UNIT III

Human Organ Systems And Bio Designs (Qualitative): Brain as a CPU system (architecture, CNS and Peripheral Nervous System, signal transmission, EEG, Robotic arms for prosthetics. Engineering solutions for Parkinson's disease). Eye as a Camera system (architecture of rod and cone cells, optical corrections, cataract, lens materials, bionic eye). Heart as a pump system (architecture, electrical signalling - ECG monitoring and heart related issues, reasons for blockages of blood vessels, design of stents, pace makers, defibrillators). Lungs as purification system (architecture, gas exchange mechanisms, spirometry, abnormal lung physiology - COPD, Ventilators, Heart-lung machine). Kidney as a filtration system (architecture, mechanism of filtration, CKD, dialysis systems).

8 Hours

UNIT IV

Nature-Bioinspired Materials And Mechanisms (Qualitative):
Echolocation (ultrasonography, sonars), Photosynthesis (photovoltaic cells, bionic leaf). Bird flying (GPS and aircrafts), Lotus leaf effect (Super hydrophobic and self-cleaning surfaces), Plant burrs (Velcro), Shark skin (Friction reducing swim suits), Kingfisher beak (Bullet train). Human Blood substitutes-hemoglobin-based oxygen carriers (HBOCs) and perflourocarbons (PFCs).

8 Hours

UNIT V

Trends In Bioengineering (Qualitative): Muscular and Skeletal Systems as scaffolds (architecture, mechanisms, bioengineering solutions for muscular dystrophy and osteoporosis), scaffolds and tissue engineering, Bioprinting techniques and materials, 3D printing of ear, bone and skin. 3D printed foods. Electrical tongue and electrical nose in food science, DNA origami and

Biocomputing, Bioimaging and Artificial Intelligence for disease diagnosis. Self- healing Bioconcrete (based on bacillus spores, calcium lactate nutrients and biomineralization processes) and Bioremediation and Biomining via microbial surface adsorption (removal of heavy metals like Lead, Cadmium, Mercury, Arsenic).
8 Hours

Text Books:

1. Biology for Engineers, Rajendra Singh C and Rathnakar Rao N, Rajendra Singh C and Rathnakar Rao N Publishing, Bengaluru, 2023.
2. Human Physiology, Stuart Fox, Krista Rompolski, McGraw-Hill eBook. 16th Edition, 2022.
3. Biology for Engineers, Thyagarajan S., Selvamurugan N., Rajesh M.P., Nazeer R.A., Thilagaraj W., Barathi S., and Jaganthan M.K., Tata McGraw-Hill, New Delhi, 2012.
4. Biology for Engineers, Arthur T. Johnson, CRC Press, Taylor and Francis, 2011.
5. Biomedical Instrumentation, Leslie Cromwell, Prentice Hall 2011.
6. Biology for Engineers, Sohini Singh and Tanu Allen, Vayu Education of India, New Delhi, 2014.
7. Biomimetics: Nature-Based Innovation, Yoseph Bar-Cohen, 1st edition, 2012, CRC Press.
8. Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies, D. Floreano and C. Mattiussi, MIT Press, 2008.
9. Bioremediation of heavy metals: bacterial participation, by C R Sunilkumar, N Geetha A C Udayashankar Lambert Academic Publishing, 2019.
10. 3D Bioprinting: Fundamentals, Principles and Applications by Ibrahim Ozbolat, Academic Press, 2016.
11. Electronic Noses and Tongues in Food Science, Maria Rodriguez Mende, Academic Press, 2016.

Web links and Video Lectures (e-Resources):

- https://nptel.ac.in/courses/121106008
- https://freevideolectures.com/course/4877/nptel-biology-engineers-other-non-biologists
- https://ocw.mit.edu/courses/20-020-introduction-to-biological-engineering-design-spring-2009
- https://ocw.mit.edu/courses/20-010j-introduction-to-bioengineering-be-010j-spring-2006

- https://www.coursera.org/courses?query=biology
- https://onlinecourses.nptel.ac.in/noc19_ge31/preview
- https://www.classcentral.com/subject/biology
- https://www.futurelearn.com/courses/biology-basic-concepts

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Group Discussion of Case studies.
- Model Making and seminar/poster presentations.
- Design of novel device/equipment like Cellulose-based water filters, Filtration system.

Course Outcomes: Upon completion of this course the student will be able to:	
CO1.	Elucidate the basic biological concepts via relevant industrial applications and case studies.
CO2.	Evaluate the principles of design and development, for exploring novel bioengineering projects.
CO3.	Corroborate the concepts of biomimetics for specific requirements.
CO4.	Think critically towards exploring innovative biobased solutions for socially relevant problems.

Universal Human Values

Contact Hours/ Week:	1L	Credits:	1.0
Total Lecture Hours:	13	CIE Marks:	50
Sub. Code:	SHS02	SEE Marks:	50

Pre-requisites: Universal Human Values (conducted during induction programme)

Course objectives: This course will enable students to:	
1	Understanding of self-exploration about themselves (human beings), family, society and nature/existence.
2	Appreciating the harmony in the human being, family, society and nature/existence
3	Strengthening holistic perception of co-existence and mutual fulfillment among the four orders of nature.

UNIT I
Understanding Harmony in the Human Being - Harmony in self: Understanding human being as a co-existence of the sentient 'I' and the material 'Body'; Understanding the needs of Self ('I') and 'Body' - happiness and physical facility; Understanding the Body as an instrument of 'I' (I being the doer, seer and enjoyer); Understanding the characteristics and activities of 'I' and harmony in 'I'.
3 Hours

UNIT II
Understanding Harmony in self and body: Understanding the harmony of 'I' with the Body: Sanyam and Health, correct appraisal of Physical needs, meaning of Prosperity in detail, Include discussions to differentiate between i) Prosperity and accumulation. ii) Ensuring health vs dealing with disease.
2 Hours

UNIT III
Understanding Harmony in the Family - Harmony in Human-Human Relationship: Understanding values in human - human relationship, meaning of Justice (nine universal values in relationships) and program for its fulfilment to ensure mutual happiness, Trust and Respect as the foundational values of relationship; Understanding the meaning of Trust, Difference between intention and competence; Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship.

3 Hours

UNIT IV

Understanding Harmony in Society and Nature: Understanding the harmony in the society (society being an extension of family)- Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals. Understanding the harmony in the Nature; Interconnectedness and mutual fulfilment among the four orders of nature- recyclability and self-regulation in nature.

2 Hours

UNIT V

Understanding Harmony in all levels of Existence: Understanding Existence as Co-existence of mutually interacting units in all-pervasive space; Holistic perception of harmony at all levels of existence. Include discussions on-human being as cause of imbalance in nature (film "Home" can be used), pollution, depletion of resources and role of technology etc.

3 Hours

TEXT BOOKS

1	Gaur, R.R. & Sangal R – 'Foundation Course in Human Values and Professional Ethics; Presenting a universal approach to value education through self-exploration', Excel Books, Bangalore, 2016, ISBN: 978-8-174-46781-2

REFERENCE BOOKS

1	Tripathi A.N.	'Human Values', New Age International Publisher, 2003, ISBN: 81-224-1426-5

Web Resource:

1. http://www.storyofstuff.com
2. https://www.youtube.com/channel/UCQxWr5QB_eZUnwxSwxXEkQw
3. https://fdp-si.aicte-india.org/8dayUHV_download.php
4. https://www.youtube.com/watch?v=8ovkLRYXIjE
5. https://www.youtube.com/watch?v=OgdNx0X923I

Course Outcomes:

Upon completion of this course the student will be able to:

CO1.	Become more aware of themselves, and their surroundings (family, society, nature)
CO2.	Become more responsible in life, and value human relationships and human society
CO3.	Have better critical ability in handling problems and in finding sustainable solutions

Evaluation Pattern:

Two Tests of 25 marks each and 45 minutes duration

SEE for 50 marks and examination duration is 90 minutes

Description	*Schedule*	*Duration (min)*	*Conducted for*	*Reduced to*
Test-1	7th Week	45	25 marks	25 marks
Test-2	14th Week	45	25 marks	25 marks
CIE			50 marks	50 marks
SEE		90	50 marks	50 marks
Total			**100 marks**	**100 marks**

Question Paper Pattern

CIE: *CIE pattern may be hybrid type with MCQs and descriptive questions.*

- *10 Marks MCQs*
- *3 descriptive questions of 5 marks each*
- *All the questions are compulsory*

SEE: *SEE pattern may also be hybrid type with MCQs and descriptive questions.*

- *20 Marks MCQs*
- *1 descriptive question of 10 marks from each of the units (total 3 questions)*

  or

  2 questions of 5 marks from each of the units (total 6 questions)
- *All the questions are compulsory*

Swift Programming
(Ability Enhancement Course-IV-1)

Contact Hours/ Week:	2P		Credits:	1.0
Total Practical Hours:	26		CIE Marks:	50
Sub. Code:	S4ISA01		SEE Marks:	50

Course objectives:
This course will enable students to:

1.	This course introduces the concept of programming for iOS.
2.	It includes iOS programming using Swift.
3.	It covers basics of Swift data types, control and loops.
4.	It also covers classes, structures, enumerations etc.

List of Problems for Laboratory

1. Write a swift program to demonstrate the different data type representation and declaring variables.
2. Write a swift program to demonstrate the concatenation of strings, variable values within the string.
3. Switches support any kind of data and a wide variety of comparison operations. Demonstrate this with swift program.
4. Dictionaries are an unordered collection. Write a swift program to find largest number and which kind of number was the largest.
5. Make a subclass of NamedShape called Circle that takes a radius and a name as arguments to its initializer. Implement an area() and a simpleDescription() method on the Circle class.
6. Define Rank enumeration to store Acc as 1 and other cards values with simpleDescription function to display message corresponding to value of Rank.
7. Demonstrate with swift program, Classes, enumerations, and structs can all adopt protocols.
8. Demonstrate with swift program to handle errors.
9. Write a swift program to demonstrate the arrays, sets and Dictionary.
10. Write a swift program to demonstrate the Early Exit.

TEXT BOOKS

1	Carlos M. Icaza	The Swift Programming Language https://carlosicaza.com/swiftbooks/SwiftLanguage.pdf

REFERENCE BOOKS

1	Learn Swift Programming URL: https://www.programiz.com/swift-programming

Course Outcomes:	
Upon completion of this course the student will be able to:	
CO 1.	**Comprehend** the basic concepts of iOS programming.
CO 2.	**Apply** the features of Swift data types to store the data.
CO 3.	**Apply** and design the class required for the applications.
CO 4.	**Develop** applications using arrays, sets and Dictionary.

Mobile Application Development
(Ability Enhancement Course-IV-2)

Contact Hours/ Week:	2P	Credits:	1
Total Practical Hours:	26	CIE Marks:	50
Sub. Code:	S4ISA02	SEE Marks:	50

Course objectives:

This course will enable students to:

1.	Explore the components and structure of mobile application development frameworks for Android.
2.	Understand fundamentals of Android operating systems.
3.	Install and configure Android application development tools Like Android Studio and Eclipse IDE.
4.	Design real time mobile applications to solve the problems.
5.	Develop user interface for mobile Application using widgets with event handling.

List of Problems for laboratory

I Introduction to

- Android Studio IDE
- The user interface
- Navigation
- Style and formatting
- Version control basics
- Managing dependencies
- Debug and profile tools

Create the applications using Android Studio for the following tasks:

1. Displaying “Welcome to Android Laboratory”.
2. Displaying different Shapes.
3. Designing Simple Calculator Application.
4. Creating an Alarm.
5. Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed.
6. Develop an Android application using controls like Button, TextView, EditText for designing a calculator having basic functionality like Addition, Subtraction, Multiplication, and Division.
7. Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules:

- Password should contain uppercase and lowercase letters.
- Password should contain letters and numbers.
- Password should contain special characters.
- Minimum length of the password (the default value is 8).

8. Write an android program to demonstrate scroll view and list view. (List view should array adapter. The adapter should use array list of companies. Each item in the list view should have company name, company address and its annual revenue.)

REFERENCES:

1.	Christopher D. Manning and Hinrich Schutze,	"Foundations of Natural Language Processing", 6th Edition, The MIT Press Cambridge, Massachusetts London, England, 2003.
2.	Daniel Jurafsky and James H. Martin	"Speech and Language Processing", Prentice Hall, 3rd Edition 2009.
3.	https://www.geeksforgeeks.org/natural-language-processing-overview	
4.	https://www.geeksforgeeks.org/python-word-similarity-using-spacy/?ref=rp	
5.	https://pub.towardsai.net/natural-language-processing-nlp-with-python-tutorial-for-beginners-1f54e610a1a0	
6.	https://www.analyticsvidhya.com/blog/2021/02/basics-of-natural-language-processing-nlp-basics/	
7.	https://towardsdatascience.com/free-hands-on-tutorials-to-get-started-in-natural-language-processing6a378e24dbfc.	

Course Outcomes:

Upon completion of this course the student will be able to:

CO1:	Design applications for Alarms, Web based applications and Location based Applications.
CO2:	Apply essential Android Programming concepts.
CO3:	Develop various Android applications related to layouts & rich uses interactive interfaces.
CO4:	Develop Android applications related to mobile related server-less database like SQLITE.
CO5:	Deploy applications to the Android marketplace for distribution.

Java Application Development for Healthcare and Agriculture
(Ability Enhancement Course-IV-3)

Contact Hours/ Week:	2P	Credits:	1.0
Total Practical Hours:	26	CIE Marks:	50
Sub. Code:	S4ISA03	SEE Marks:	50

Course objectives:	
This course will enable students to:	
1.	This course introduces the concept of developing an application using any language.
2.	It includes both a front-end and a back-end design concept.

List of Applications
1. Blockchain based health record system.
2. Health related problem searching and creating database.
3. Suggesting medicine based on symptoms.
4. Identifying the side effects of a medicine.
5. Impact of sugar consumption on the health.
6. A study on patient satisfaction in health screening department in a multi-specialty hospital.
7. Crop planning and management system.
8. Farm operation management.
9. Accounting for farms.

Course Outcomes:	
Upon completion of this course the student will be able to:	
CO1:	**Analyze** and **apply** JAVA programming language for application development.
CO2:	**Design** the application for providing solution to health care related problem.
CO3:	**Design** the application for agriculture based problem.
CO4:	**Demonstrate** the features of the application developed.

Natural Language Processing
(Ability Enhancement Course-IV-4)

Contact Hours/ Week:	: 2P	Credits:	1
Total Practical Hours:	: 26	CIE Marks:	50
Sub. Code:	: S4ISA04	SEE Marks:	50

Course objectives:	
This course will enable students to:	
1.	Describe the concepts of morphology, syntax, semantics, discourse & pragmatics of natural language.
2.	Describe the various approaches to understand syntax and semantics in NLP.
3.	Use and apply language related tools that are available to efficiently study and analyze large collections of text data.
4.	Demonstrate understanding of the relationship between NLP and statistics & machine learning.
5.	Familiarize with various NLP software libraries and data sets publicly available.
6.	Illustrate how language technology relies on formal models to capture Knowledge.

Introduction to Natural language processing

- Introduction to Python.
- Introduction to Natural Language Toolkit (NLTK).
- Python quick overview.
- Lexical analysis: Word and text tokenizer; – n-gram and collocations.
- NLTK corpora (Publicly available).

List of Programs:

Write programs for the following (Use input the English language text sample input text files may be copied from newspaper/ Online articles):

1. Create a small text file, and write a program to read it and print it with a line number at the start of each line. (Make sure you don't introduce an extra blank line between each line).
2. Write a function named word_freq() that takes a word as input and compute the frequency of the occurrence of the word in that section of the corpus. Test your result with the help of a frequency distribution library function (FreqDist ()) in NLTK.
3. Define a function percent (word, text) that calculates how often a given word occurs in a text and expresses the result as a percentage.
4. Find all the four-letter words from the given text file. Show these words in decreasing order of frequency.
5. Implementing simple problems related to word disambiguation.

6. Implementation of tokenization of text.
7. Implementation of stop word removal, Stemming of text and Lemmatization.
8. Implementation of n-gram model.
9. Implementation of PoS tagging.
10. Implementation and demonstration of Named entity recognition.

REFERENCES:

1.	Christopher D. Manning and Hinrich Schutze,	"Foundations of Natural Language Processing", 6th Edition, The MIT Press Cambridge, Massachusetts London, England, 2003.
2.	Daniel Jurafsky and James H. Martin	"Speech and Language Processing", Prentice Hall, 3rd Edition 2009.
3.	https://www.geeksforgeeks.org/natural-language-processing-overview	
4.	https://www.geeksforgeeks.org/python-word-similarity-using-spacy/?ref=rp	
5.	https://pub.towardsai.net/natural-language-processing-nlp-with-python-tutorial-for-beginners-1f54e610a1a0	
6.	https://www.analyticsvidhya.com/blog/2021/02/basics-of-natural-language-processing-nlp-basics/	
7.	https://towardsdatascience.com/free-hands-on-tutorials-to-get-started-in-natural-language-processing6a378e24dbfc.	

Course Outcomes:

Upon completion of this course the student will be able to:

CO1.	**Identify** various linguistic and statistical features relevant to the basic NLP task, namely, spelling correction, morphological analysis.
CO2.	**Apply** and compare natural language processing with manual and automated approaches.
CO3.	**Apply** Part-of-speech (POS) tagging for a given natural language and suitable modeling technique based on the structure.
CO4.	**Apply** the state of the art algorithms and techniques for text-based processing of natural language with respect to morphology.