RashtreeyaSikshanaSamithi Trust
R.V.COLLEGE OF ENGINEERING
(Autonomous Institution Affiliated to VTU, Belgaum)
R.V. Vidyaniketan Post, Mysore Road
Bangalore – 560 059

Scheme &Syllabus

III & IV Semester B.E.
INFORMATION SCIENCE & ENGINEERING

(2012 Scheme)

R. V. College of Engineering, Bangalore – 59.

(An Autonomous Institution affiliated to VTU, Belgaum)

Department of Information Science & Engineering

THIRD SEMESTER								
Sl. No	**Course Code**	**Course**	**BoS**	**Credit Allocation***				**Total Credits**
				L	**T**	**P**	**S**	
1	12MA31	Applied Mathematics-III	Sc	3	1	0	0	4
2	12EM32	Engineering Materials	ME	3	0	0	0	3
3	12IS/CS33	Data Structures in C(Theory & Practice)	ISE	3	0	1	1	5
4	12IS/TE/EE/CS/IT34	Digital Logic Design (Theory & Practice)	TE	3	0	1	1	5
5	12IS35	Object Oriented Programming with C++ (Theory & Practice)	ISE	3	0	1	1	5
6	12IS/CS36	Discrete Mathematical Structures	CSE	3	1	0	1	5
7	12DMA37	Bridge Course Mathematics – I**	Sc					
								27
		No. of Hrs.		**18**	**04**	**06**	**16**	**44**

FOURTH SEMESTER								
Sl. No.	**Course Code**	**Course**	**BoS**	**Credit Allocation***				**Total Credits**
				L	**T**	**P**	**S**	
1	12MA41	Applied Mathematics IV	Sc	3	1	0	0	4
2	12EB42	Environmental Science and Biology for Engineers	Sc	3	0	0	1	4
3	12IS/CS43	Theory of Computation	ISE	3	0	0	1	4
4	12IS44	Computer Organization and Architecture	ISE	3	0	0	0	3
5	12IS/CS45	Design and Analysis of Algorithms (Theory & Practice)	CSE	3	0	1	1	5
6	12IS46	Operating Systems	ISE	3	0	0	0	3
7	12HSS47	Innovation and Social skills	HSS	0	0	1	0	1
8	12DMA48	Bridge Course Mathematics-II**	Sc					
9	12IS49	Unix Systems Programming (Theory & Practice)	ISE	3	0	1	1	5
								29
		No. of Hrs.		**21**	**02**	**06**	**16**	**45**

** Mandatory Audit course for Lateral entry (Diploma students)

*L – Lecture, T – Tutorial, P – Practical, S – Self Study

Semester - III
Applied Mathematics III
(ALL BRANCHES)

Course Code:	**12MA31**	**CIE Marks:**	**100**
L:T:P:S	**3:1:0:0**	**SEE Marks:**	**100**
Credits:	**04**	**SEE :**	**3 Hrs**

Course Learning Objectives:

- The student should be able to analyze periodic phenomena using concept of Fourier series.
- Understand the basics of matrix theory and its applications for finding solution of system of linear equations.
- Finding the approximate solutions using numerical methods, for problems which do not have analytical solutions.
- Approximating functional values with different curves.
- Optimizing real functional with various applications.

UNIT – I **08 Hrs**

Fourier series and Fourier Transforms:

Introduction, periodic functions, Even and odd functions, properties. Special waveforms - Square wave, half wave rectifier, saw-tooth wave and triangular wave. Euler's formula for Fourier series, Fourier series for functions of period 2L (particular cases), Dirichlet's conditions - problems. Half Range Fourier series- Construction of Half range cosine and sine series, Complex form of Fourier series. Complex Fourier Transforms –Properties & simple problems.

UNIT – II **07 Hrs**

Matrices and Linear Equations:

Elementary transformation, rank of matrix by using Echelon form, consistency of system of linear equations and solutions, solution of system of linear equations using Gauss elimination method, Gauss Jordan method, Gauss Seidel method, Eigenvalues and Eigenvectors, finding largest eigenvalue by using Power method.

UNIT – III **07 Hrs**

Curve Fitting and Interpolation:

Method of Least squares - fitting of the curves of the form $y = ax + b$, $y = ae^{bx}$, $y = ax^b$ and $y = ax^2 + bx + c$, Correlation and Regression analysis. Finite differences-forward and backward differences, Interpolation-Newton's forward and backward interpolation formulae, Lagrange's interpolation formula.

UNIT – IV **07 Hrs**

Numerical Methods:

Numerical integration– Simpson's rules, Weddle's rule and Gaussian quadrature (two point & three point formula). Numerical methods for first order ODE – Single step & Multistep methods-Taylor's series method, Runge-Kutta fourth order method, Adam-Bash forth's method, BVP for ODE – Shooting methods for second order ODE (All methods without proof).

UNIT – V **07 Hrs**

Calculus of Variation: Introduction, Variation of functions and functional, extremal of a functional, variational problem, Euler's equation and special cases. Examples - Geodesics, Hanging cable, and Brachistochrome problem.

Course outcomes:

At the end of thiscourse the student will be able to:

- Apply knowledge of linear algebra for finding the solution of system of linear equations.
- Analyze and interpret physical phenomena which are periodic in nature by applying Fourier series.
- Solve Algebraic and transcendental equations using effective numerical methods.

Reference Books:

1. B.S. Grewal - Higher Engineering Mathematics, Khanna Publishers, 40th Edition, 2007, ISBN: 81-7409-195-5, Chapters 2, 10, 24, 28, 29, 31, 34.
2. N.P Bali & Manish Goyal - A Text Book of Engineering Mathematics, Lakshmi Publications, 7thEdition, 2010, ISBN: 978-81-7008-992-6, Chapters: 3(3.34-3.40,3.46, 3.47), 10 (10.1-10.7-10.10), 2 (2.24 -2.26).
3. Erwin Kreyszig - Advanced Engineering Mathematics, John Wiley & Sons, 9th Edition, 2007, ISBN: 978-81-265-3135-6, Chapters: 6, 7.1, 7.2,10(10.1-10.5,10.9-10.11),17, 18,19.
4. Murray R Spiegel - Theory & problems of Fourier Analysis with Applications to Boundary Value Problems, Schaum's Outline Series.

Scheme of Continuous Internal Evaluation:

CIE consists of Three Tests each for 45 marks (15 marks for Quiz + 30 marks for descriptive) out of which best of two will be considered. In addition there will be one seminar on new topics / model presentation etc. for 10 marks.

Scheme of Semester End Examination:

The question paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions (descriptive, analytical, problems or/and design) carrying 16 marks each. All five from Part B will have internal choice and one of the two have to be answered compulsorily.

Engineering Materials
(Common to all Branches)

Course Code:	**12EM3S**	**CIE Marks:**	**100**
L:T:P:S	**3:0:0:0**	**SEE Marks:**	**100**
Credits:	**03**	**SEE :**	**3 Hrs**

Course Learning Objectives:

By the end of the course the student should be able to:

1. Compare materials based on their properties.
2. Identify appropriate materials for specific engineering applications.
3. Identify the phases and structure – property relations in alloys based on the phase diagrams.
4. Apply Define and differentiate thermodynamic work and heat.
5. Select materials and process parameters for Flexible Electronics Technology.
6. Identify appropriate nano materials for engineering applications and their characterization.
7. Identify materials and property requirements for advanced engineering applications.

UNIT – I **06 Hrs**

Introduction: Classification of Materials - Metals, Ceramics, Polymers, composites, Advanced Materials- semiconductors, biomaterials, smart materials, nanostructured materials and their applications.

Material Properties: Mechanical properties, thermal properties – Heat capacity, CTE, thermal conductivity, Electrical and Electronic conductivity, Magnetic properties – dia, para, ferro, ferri, antiferro, domains and hysteresis. Optical properties -Luminescence and photoconductivity.

UNIT – II **08 Hrs**

Ferrous Materials and Alloys - Binary phase diagrams, Phase Rule, Lever Rule, Solidification, Nucleation and Grain Growth.

Cast Iron,Chromium steels, Nickel steels, Silicon Steels, Tungsten and Molybdenum Steels & Stainless Steels; Tool Steels, structural steels, Corrosion and Heat Treatment

Non-ferrous Materials and Alloys: Aluminum, Copper and Titanium, their alloys, properties and applications.

UNIT – III **08 Hrs**

Overview of Flexible Electronics Technology: History of Flexible Electronics, Materials for Flexible Electronics , Fabrication Technology for Flexible Electronics Fabrication on Sheets by Batch Processing, Fabrication on Web by Roll-to-Roll Processing , Additive Printing, Low-temperature Amorphous and NanocrystallineSilicon Materials, Low-temperature Dielectrics, Low-temperature Thin-film Transistor Devices

Ceramic Materials: Definition, Classification of Ceramic Materials, Processing Methods, Properties and Industrial, Medical and Commercial Applications**.**

Polymers: Definition, Classification of Polymers, Properties and their applications, intrinsically conductive material**.**

UNIT – IV **08 Hrs**

Composites: Types of Matrix Materials and Reinforcements, Selection of Composites, Properties, Applications, Rule of Mixture for density, elastic modulus and tensile strength.

Nanomaterials - Definition, classification and synthesis – physical and chemical processes, Characterization of nanomaterials – Electron microscope, X-Ray Diffraction, particle size analyzer**.**

UNIT – V **06 Hrs**

Advanced materials for - Construction Applications, Biomedical applications, High temperature Applications, Sensors and Actuators - Shape Memory Alloys and Composites, Thin films and coatings.

Course Outcome:

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

1. Classify materials based on properties
2. Compute the properties of composites based on the properties of the constituents
3. Draw Binary phase diagrams and identify the phases
4. Identify characterization techniques for nanomaterials, thin films, flexible electronics, biomedical applications, high temperature applications, sensors and actuators

Reference Books:

1. William D. Callister; "Materials Science & Engineering- An Introduction"; Wiley India Pvt. Ltd.; 6th Edition; 2006; New Delhi; ISBN:9814-12-669-1; 1,4,6,7,8,9.
2. Fred W. Billmeyer, Jr; " Text Book Of Polymer Science"; Wiley-Interscience Publication; 2nd Edition; 1984; ISBN:0-471-82834-3; 8.
3. Donald R. Askland, Pradeep P. Phule, "Essentials of Materials Science and Engineering", Thomas Canada Learning INDIA EDITION, ISBN:81-315-0233-3.
4. William Smith,"Foundation of Materials Science and Engineering", 3rdEdition, McGraw Hill, 1997. ISBN:9780073529240.
5. Flexible Electronics: Materials and Applications, William S. Wong and Alberto Salleo, eds.ISBN 978-0-387-74362-2, 200.9

Scheme of Continuous Internal Evaluation:

CIE consists of Three Tests each for 45 marks (15 marks for Quiz + 30 marks for descriptive) out of which best of two will be considered. In addition there will be one seminar on new topics / model presentation etc. for 10 marks.

Scheme of Semester End Examination:

The question paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions (descriptive, analytical, problems or/and design) carrying 16 marks each. All five from Part B will have internal choice and one of the two have to be answered compulsorily.

Data Structures in C
(Theory and Practice)

Course Code:	**12IS/CS 33**	**CIE Marks:**	**100 + 50**
L:T:P:S	**3:0:1:1**	**SEE Marks:**	**100 + 50**
Credits:	**05**	**SEE :**	**03+03 Hrs**

Course Learning Objectives - CLO:

1. To learn the fundamental data structures andidentify data structuring strategies that are appropriate to a given contextual problem and able to design, develop, test and debug in C language considering appropriate data structure.
2. Illustrate and implement how data types such as stack, queue and linked list can be implemented to manage the memory using static and dynamic allocations.
3. Understand and distinguish the conceptual and applicative differences in trees, binary trees, and binary search trees. Implement binary tree traversals and operations on binary search trees to design applications like expression trees. Understand Hashing and resolve hash clashes
4. To create and use appropriate data structures in C programs for solving real life problems.

UNIT – I **07 Hrs**

Introduction to Data Structure: Types of Data Structures, Linear &non linear Data Structures.
Stacks: Stack definitions & concepts, Representing stacks in C, Operations on stacks, Applications of Stacks, Infix to Postfix, Postfix expression evaluation
Recursion: Introduction to Recursion, Factorial function, Binary search, Towers of Hanoi problem, Role of the stack during execution.

UNIT – II **07 Hrs**

Queue: Representation of queue, operations, circular queues, Implementation of Priority queues & Binary Heap
Dynamic Memory Allocation: malloc(), calloc(),free(), realloc()

UNIT – III **07 Hrs**

Lists: Inserting and removing nodes from a list, linked implementation of Stacks, Queues. getnode and freenode operations, Example of list operations, Array implementation, limitations of the array implementation, Linked list using dynamic variables and its operations.

UNIT – IV **07 Hrs**

Other List Structures: Circular list, stack and queue as a circular list, primitive operations on circular list, addition of long positive integers using circular list, Header nodes, Doubly linked list and its operations.

UNIT – V **08 Hrs**

Hashing: Resolving hash clashes by open addressing, deleting items from hash table, separate chaining.
Trees: Binary Search Tree (BST), Insertion and Deletion in BST, Expression trees, Infix, Postfix, and Prefix Traversals, Splay tree, Tries.

Laboratory Component

1. Use Stack operations to do the following:
   i) Assign to a variable name Y the value of the third element from the top of the stack and keep the stack undisturbed.
   ii) Given an arbitrary integer n pop out the top n elements. A message should be displayed if an unusual condition is encountered.
   iii) Assign to a variable name Y the value of the third element from the bottom of the stack and keep the stack undisturbed.
   (Hint: you may use a temporary stack)

2. Write a program to determine if an input character string is of the form XaY. X is a string of arbitrary length using only the characters from A and B. For example, X may be ABBAB. Y is a string which is the reverse of X. Thus for the string X given above, Y is BABBA. A is any arbitrary character which is not A or B. Given ABAAcABAB the program should display a message that this string is invalid. For the string ABBABCBABBA the program should write a message that it is valid. Use appropriate data structure.

3. Write a C program that parses Infix arithmetic expressions to Postfix arithmetic expressions using a Stack.

4. Write a C program to Build Binary Heap to simulate Priority queue.

5. Write a C program to simulate the working of Messaging System in which a message is placed in a Queue by a Message Sender, a message is removed from the queue by a Message Receiver, which can also display the contents of the Queue.

6. Write a C program to accept 2 singly linked lists & print the elements which are common in both the lists.

7. Implement working of lift using appropriate data structure.

8. Consider a linked list with n integers. Each node of the list is numbered from 1 to n. Develop a program using 'C' language to split this list into 4 lists so that:
   a. First list contains nodes numbered 1,5,9,13,_,_,_
   b. Second list contains nodes numbered 2,6,10,14,_,_
   c. Third list contains nodes numbered 3,7,11,15,_,_
   d. Fourth list contains nodes numbered 4,8,12,16,_

9. Implement a program to multiply two polynomials using circular linked list.

10. Design a doubly linked list to represent sparse matrix. Each node in the list can have the row and column index of the matrix element and the value of the element.

11. Write a C program to implement Hashing using Open Addressing.

12. Write a C program to create Binary Tree and provide insertion and deletion operations and to traverse the tree using In-order, Preorder and Post order (recursively)

13. Given a String representing a parentheses-free infix arithmetic expression, implement a program to place it in a tree in the infix form. Assume that a variable name is a single letter. Traverse the tree to produce an equivalent postfix and prefix expression string.
14. Implement a simple Dictionary using Trees.

Course outcome:

At the end of the course, the student will be able to -

1. Exhibit program design and implementation competence through the choice of appropriate data structure.
2. Build new data structures based on the need of the application.
3. Understand how choice of data structures influences the performance of programs.
4. Visualize the need of appropriate data structures in solving real-life problems

Reference Books:

1. YedidyahLangsam Moshe J. Augenstein and Aaron M. Tenenbaum; Data Structures using C and C++, PHI/Pearson, 2nd Edition, 2009.
2. Computer Science A Structured Programming Approach Using C by Behrouz A. Forouzan and Richard F. Gilberg, 3rd Edition, Thomson, 2009.
3. Data Structures and Program Design in C++, R.Kruse, C.L Tondo and B.Leung, 2nd Edition, Pearson Education, 2009.
4. Introduction to Data Structures in C, Ashok N Kamthane, Pearson Education, 1st Edition, 2009.

General Guidelines

1. GOTO statements are not allowed
2. No global declarations allowed
3. Prototype for each user-defined-function must be provided before main
4. main should be the first function in any program
5. Students are encouraged to use user-defined-header files
6. Programs must be indented appropriately

Scheme of Continuous Internal Evaluation (CIE):

CIE consists of Three Tests each for 40 marks (15 marks for Quiz + 25 marks for descriptive) out of which best two will be considered. In addition 20 marks to be earned through self study component on emerging topics.

Scheme of Continuous Internal Evaluation for Practical's:

Cumulative Continuous Evaluation is done for a total of 50 marks. Out of that 30 marks are for execution of programs and record writing. Internal TEST Evaluation is for 20 marks.

Scheme of Semester End Examination (SEE):

The questions paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions one from each unit (descriptive, analytical, problems or/ and design) carrying 16 marks each. All five questions from Part B will have internal choice – one of the two questions has to be answered compulsorily.

Scheme of Semester End Examination for Practical's:

Students should select a program from the LOT of 14 programs. The student will be evaluated for 40 marks based on the Write-up, Execution, and Results. Viva Voce will be for 10 Marks.

Digital Logic Design

(Theory and Practice)

Course Code :12IS/TE/EE/CS/IT 34 **CIE Marks: 150**
L:T:P:S 3:0:1:1 **SEE Marks: 150**
Total credits: 05 **SEE hours: 3+3**

Course Learning Objectives (CLO):

1. To optimize logic expressions using Karnaugh map and Tabular method
2. To simplify Boolean equation and design combinational circuits with optimal gates.
3. To Analyze the working principles of Flip-Flops and design asynchronous sequential circuits
4. To design simple synchronous digital circuits based on finite state machine algorithm
5. To design, simulate and implement digital systems using HDL

UNIT – I **07 Hrs**

Simplification of Boolean Expressions: Formulation of the Simplification Problem, Prime Implicants and Irredundant Disjunctive Expressions, Prime Implicates and Irredundant Conjunctive Expressions, Karnaughs Map- Using Karnaugh Maps to obtain minimal Expressions for Complete Boolean functions, Minimal Expressions of Incomplete Boolean Expressions, The Quine MC-Cluskey Method of generating Prime implicants and Prime implicates, Prime-Implicant / Prime-Implicate Tables and Irredundant expressions, Prime-Implicant / Prime-Implicate Table Reductions, VEM Technique (up to 4 variables), Binary Adders and Subtractors, Decimal Adders.

UNIT – II **07 Hrs**

Logic Design with MSI Components and Programmable Logic Devices (PLD's): Comparators, Decoders, Encoders, Parity Generators and Parity Checking Circuits, Multiplexers, Programmable Logic Devices, Programmable Read-only memories, Programmable Logic Arrays, Programmable Array Logic and design of combinational circuits using PLD's.

UNIT – III **08 Hrs**

Flip-Flops and Applications: The Basic Bistable Elements, Latches, Timing Considerations, Master-Slave Flip-Flops (Pulse-triggerred Flip-Flops), Edge – Triggerred Flip-Flops, Characteristics Equations, Registers, Counters, Design of Synchronous and asynchronous Counters.

UNIT – IV **08 Hrs**

Synchronous Sequential Networks: Structure and operation of Clocked synchronous Sequential Networks, Analysis of Clocked Synchronous Sequential Networks, Modeling clocked synchronous sequential network behavior, State Table Deduction, The State Assignment, Completing the design of clocked synchronous sequential networks

UNIT – V **06 Hrs**

Logic Families: Transistor – Transistor Logic (TTL), Emitter – Coupled Logic (ECL), The MOS Field Effect Transistor, NMOS and PMOS Logic, CMOS Logic, Interface between logic families, Tri-state buffer, RAM and ROM memory, Static and dynamic hazards in logic circuits.

Laboratory Component
PART A

1. Realization of Parallel adder / subtractor using IC 7483
2. Design and Realization of adder, subtractor using IC 74153 and binary to gray code conversion using IC 74139
3. Design and realization of One/Two bit comparators using basic gates. Realization of 4 bit comparators using IC 7485.
4. Realization of decoder, encoders and priority encoders
5. Realization and verification of SR and JK flip-flops using universal gate. Realization of Master-Slave flip-flop using IC7476.
6. Design of programmable counters using IC74192 & IC74193
7. Realization of ring counter and Johnson counter.
8. Design and verification of Parity generators and parity checkers.

PART B

The students are required to design any one digital system using the concepts learnt in PART A. The designed circuit has to be realized using discrete hardware components and implement on FPGA using HDL

1. Addition of two numbers whose sum is less than 9
2. Design a Stop Clock to display from 0 to 9 Sec.
3. Design a Stop Clock to display from 1 to 9 min.
4. Design a Circuit that will display random numbers from 0 to 9.
5. Design a circuit that will transmit 4 bit of information serially / over a single channel.
6. Sequence Generator
7. Switch debouncer
8. Programmable Signal Generator
9. 4 bit by 3 bit binary multiplier
10. Data serializer
11. Design of parity generator and checker using multiplexer
12. Design a digital system to control a dc motor using decoder
13. Design a digital system to generate carry, overflow and auxiliary carry for an 8 bit addition and subtraction using suitable IC`s
14. Design a 2 bit comparator using PAL
15. Design a driver circuits with current rating

Course Outcome :

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

1. Simplify Boolean expressions and implement optimal Logic circuits
2. Design and implement combinational and sequential digital systems
3. Design and implement synchronous digital systems using state machines
4. Conceptualize design a digital system using programmable logic arrays
5. Design and implement digital circuits using HDL on FPGA

Reference Books:

1. Donald D.Givone, "Digital Principles and Design", Tata McGraw-Hill, 2002.
2. Stephen Brown, "Fundamentals of Digital Logic Design with Verilog", Tata McGraw Hill, 2nd Edition, 2008.
3. M Morris Mano, Michael D.Ciletti, "Digital Design", Pearson, 4th edition, ISBN-978-81-317-1450-8.
4. Samir Palnitkar "Verilog HDL A Guide to Digital Design and Synthesis" Pearson Education Asia, 2nd Edition, ISBN : 81-7758-918-0

Scheme of Continuous Internal Evaluation (CIE):

CIE consists of Three Tests each for 40 marks (15 marks for Quiz + 25 marks for descriptive) out of which best two will be considered. In addition 20 marks to be earned through self study component on emerging topics.

Scheme of Continuous Internal Evaluation for Practical's:

Cumulative Continuous Evaluation is done for a total of 50 marks. Out of that 30 marks are for execution of programs and record writing. Internal TEST Evaluation is for 20 marks.

Scheme of Semester End Examination (SEE):

The questions paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions one from each unit (descriptive, analytical, problems or/ and design) carrying 16 marks each. All five questions from Part B will have internal choice – one of the two questions has to be answered compulsorily.

Scheme of Semester End Examination for Practical's:

Students should select a program from the list of 8 programs. The student will be evaluated for 40 marks based on the Write-up, Execution, and Results. Viva Voce will be for 10 Marks.

Object Oriented Programming with C++
(Theory and Practice)

Course Code : 12IS35	**CIE Marks : 100+50**
L:T:P:S :3:0:1:1	**SEE Marks : 100+50**
Credits : 05	**Exam Hours : 03+03 Hours**

Course Learning Objectives (CLO):

1. To comprehend the Object Oriented Programming features in C++.
2. To apply object oriented solutions in various implementation issues of software Systems.
3. To analyze the design and implement programs through Object Oriented Approach.
4. To Analyze the role of Object Oriented Approach in design and development of good quality software systems

UNIT – I **08 Hrs**

Object Oriented Design: The Software Lifecycle, CRC Cards, Cohesion, Coupling, Relationship Between Classes, Implementing Associations, Case Study: Printing an invoice.
The Unified Modeling Language: Graphical Notations, Use Cases, UML Class Diagrams, Sequence Diagrams, State Diagrams, Case Study: A Voice Mail System.
An Introduction to Design Patterns: Iterators, The Pattern Concept, The Adapter Pattern, The Template Method Pattern, Function Objects and the Strategy Pattern, The Composite Pattern, Case Study: Putting Patterns to Work.

UNIT – II **07 Hrs**

Functions: Procedures, Reference Parameters, Variable Scope and Global Variables, Stepwise Refinement, From Pseudocode to Code.
Objects and Classes: Constructing Objects, Using Objects, Real-Life Objects, Displaying Graphical Shapes, Graphics Structures, Choosing a Coordinate System, Getting Input from the Graphics window, Comparing Visual and Numerical Information. Discovering Classes, Interfaces, Encapsulation, Member Functions, Default Constructors, Constructors with Parameters, Accessing Data Fields, Comparing Member Functions with Nonmember Functions.
Inheritance: Deriving Classes, Calling the Base-Class Constructor, Calling the Base-Class Member Functions, Polymorphism.

UNIT – III **07 Hrs**

Operator Overloading: Operator Overloading, Case Study: Fractional Numbers, Overloading Simple Arithmetic Operators, Overloading Comparison Operators, Overloading Input and Output, Overloading Increment and Decrement Operators, Overloading Assignment Operators, Overloading Conversion Operators, Overloading Subscript Operator, Overloading Function Call Operator, Case Study: Matrices.
Memory Management: Categories of Memory, Common Memory Errors, Constructors, Destructors, Reference Counting, Case Study: Matrices.

UNIT – IV **07 Hrs**

Exception Handling: Handling Exceptional Situations, Alternative Mechanisms for Handling Exceptions, Exceptions, Case Study: Matrices.
Streams: Reading and writing Text files, The Inheritance Hierarchy of Stream classes, String Streams, Command Line Arguments, Random Access.
Polymorphism: Encapsulation, Name Class Inheritance Hierarchies, Polymorphic Variables, Virtual and Nonvirtual Overriding, Pure Virtual Member Functions, Obtaining Run-Time Typing Information, Slicing and Polymorphism, Multiple Inheritance, Software Frameworks.

UNIT – V 07 Hrs

Template Functions and Classes: Template Functions, compile-time Polymorphism, Template Classes, Template Linked List, Nontype Template Arguments, Setting Behaviour Using Template Arguments, Case Study: Matrices.

The Standard Template Library-Containers: The STL, The Fundamental Containers, The Stack and Queue Adapters, Sets, Maps, Case Study: Dijkstra's Shortest Path Algorithm.

The Standard Template Library - Iterators and Algorithms: Loose Coupling Produces a Strong Library, Iterators, Functions, Generators and Predicates, Generic Algorithms, Inserters, Stream Iterators, Standard Function Objects, Predicates, and Binders, Case Study: File Merge Sort.

Laboratory Component

1 Implement the following requirement: An electricity board charges the following rates to domestic users to discourage large conceptions of energy.

0 - 100 units : Rs 1.50 per unit
101 - 200 units : Rs 1.80 per unit
Beyond 200 units: Rs 2.50 per unit

All users are charged a minimum of Rs 50. If the total amount is more than Rs 300 then an additional surcharge of 15% is added. The C++ program must read the names of users, number of units consumed and display the calculated charges.

2 Design and implement a class RATIONAL which represents a rational number. Each rational number should be represented as two integers, the numerator and the denominator. A default constructor should initialize each declared rational number to 0, which could be 0/1. There should be constructors with 1 and with 2 parameters, to give a/1 and a/b. The constructors and other functions should prevent a 0 denominator and should prevent a negative denominator. Provide a private member function reduce ( ) to reduce a fraction to lowest terms (by dividing out the greatest common divisor). Implement member functions to add, subtract, multiply and divide two rational numbers and display the results.

3 Design and implement a class STUDENT with attributes like: roll number, name, 3 tests marks. Implement member functions

a) to read student data like name and test marks,
b) to compute average marks (considering best two out of three test marks) and
c) to display the student information.

Declare an array of STUDENT objects in the main function, use static data member to generate unique student roll number.

4 Write a C++ program to serve the following requirements:

a) Create a class OCTAL, which has the characteristics of an octal number. Implement member functions to convert a given decimal number to octal format and octal format back to decimal format.
b) Create a class HEXA, which has the characteristics of hexadecimal number. Implement member functions to convert a given decimal number to hexadecimal format and hexa format back to decimal format.

Implement a friend function to add the octal number and hexadecimal number and display the result in decimal format. Add constructors and member functions to read, display data in both the classes.

5 Design and implement a C++ class POLYNOMIAL. The internal representation of a POLYNOMIAL is an array of terms. Each term contains a coefficient and an exponent, e.g., the term $2x^4$ has the coefficient 2 and the exponent 4. Implement a class containing

constructors and the following capabilities:

a) Overload the addition operator (+) to add two polynomials
b) Overload the assignment operator to assign one polynomial to another
c) Overload the multiplication operator (*) to multiple two polynomials
d) Overload the >> operator to enable input through in.
e) Overload the << operator to enable output throughout.
f) Member function to compute value of the polynomial, given the value of x.

6 A company pays its employees weekly. The employees are of three types.

a) Salaried employees are paid a fixed weekly salary regardless of the number of hours worked.
b) Commission employees are paid a percentage of their sales and
c) Base_salary_plus_commission employees receive a base salary plus percentage of their sales.

For the current pay period, the company has decided a reward for base_salary_plus_commission employees by adding 10 percent to their base salaries. The company wants to implement a C++ program that performs its payroll calculations polymorphically. Design and implement the program using inheritance. Implement the methods to read and write the employee details (like name, emp_id, salary, etc), to compute the employee‘s gross salary.

7 Design and implement a C++ program to create an abstract class - SHAPE to represent any shape in general. The class should have two pure virtual functions to read dimensions and to compute the area. Create three derived classes - CIRCLE, RECTANGLE, and SQUARE by inheriting the features of class SHAPE. Implement the functions to read and compute the area. Add constructors, method to display the results as required. (Assume appropriate attributes).

8 Write a C++ program for custom exception handling.

i. Implement a function to compute factorial of a given number.
ii. Create a class “InvalidDataException” that contains the details about the exception – “Invalid data: negative number entered”
iii. In the main function, accept a number from the user and throw an exception of type “InvalidDataException” if the entered number is a negative number, else call the factorial function to compute the result.
iv. Handle the exception.

9 Write a C++ program using generic class to implement queue of integers, floating point numbers and strings. Support the queue operations like insert, delete and display in the queue class

10 Write a C++ program to create a vector of integers. Copy the vector contents into a list, sort the contents, then copy selected items into another vector (like elements less than 10 etc)

11. The history teacher at your school needs help in grading a true/false test. The students' IDs and test answers are stored in a text file. The first entry in the file contains answers to the test in the form:

TFFTFFTTTFFTFTFTFTT

Every other entry in the file is the studentID, followed by a blank, followed by the student's responses. For example, the entry:

ABC54301 TFTFTFTT TFTFTFFTTFT

Indicated that the student ID is ABC 54301 and the answer to question 1 is true, the answer to question 2 is false, and so on. This student did not answer question 9. The exam has 20 questions,

and the class has more than 150 students. Each correct answer is awarded two points, each wrong answer gets one point deducted and no answer gets zero points. Design and implement a C++ program that processes the test data. The output should be the student's ID, followed by the answers, followed by the test score, followed by the test grade. Assume the standard grade scale (90-100 -- a) (80-89 -- b) etc.

12. Design and implement C++ program that acts as a Simulated Company with following requirements: Simulate a small manufacturing company. The resulting application must enable the user to take out a loan, purchase a machine, and over a series of monthly production runs, follow the performance of their company

13. In C++ there is no check on an array index out of bounds. However, during program execution, an array index out of bounds can cause serious problems. Also, in C++ the array index starts at zero. Design the class myArray that solves the array index out of bound problem, and also allows the user to begin the array index starting at any integer, positive or negative. Every object of type myArray is an array of type int. During execution, when accessing an array component, if the index is out of bounds, the program must terminate with an appropriate error message. Consider the following statements:

```
myArray<int> list(5);
myArray<int>myList(2, 13);
myArray<int>yourList(-5,9);
```

The statement in line 1 declares a list to be an array of 5 elements, the element type is int, and the elements start from list[0] .... list [4]. Line 2 declares a list of 11 elements, starting at myList[2] ... myList[12]. Line 3 declares a list of 14 elements, starting at myList[-5] ... myList[8]. Write a program to test your class.

14. Design and implement C++ program using inheritance for the following requirement:
    Part 1: A point in the x-y plane is represented by its x-coordinate and y-coordinate. Design a class, pointType, that can store and process a point in the x-y plane. You should then perform operations on the point, such as setting the coordinates of the point, printing the coordinates and returns the x/y coordinate.
    Part 2: Every circle has a center and a radius. Given the radius, we can determine the circle's area and circumference. Given the center, we can determine its position in the x-y plane. The center of the circle is a point in the x-y plane. Design a class, circleType, that can store the radius and center of a circle. Because the center is in the x-y plane, you must derive the class circleType from pointType. You should be able to perform the usual operations on the circle, such as setting the radius, printing the radius, calculation and printing the area and circumference, and carrying out the usual operations on the center.
    Write a program to test various operations on a circle

Course Outcome:

1. The student will be able to Exhibit program design and implementation competence through the choice of appropriate object oriented concept.
2. The student will be able to design and analyze the Applications using Object Oriented Approach.
3. The student will be able to envision the solutions for real-time problems using Object Oriented concepts.
4. The student will be able to understand how choice of object orientation influences the performance of programs.

Reference Books:

1. Big C++, Cay S. Horstmann, Timothy Budd, Wiley India (P.) Ltd, 1st Edition, 2009, ISBN:

9788126509201.
2. The Complete Reference C++, Herbert Schildt, McGrawHill, 4th Edition, 2011, ISBN: 9780070532465.
3. C++ How to Program, Paul Deitel and Harvey Deitel, Prentice Hall, 8th Edition, 2012, ISBN: 9780132990448.
4. Object Oriented Analysis & Design With Application , Robert A Maksimchuk, Grady Booch, Pearson Education Limited, 3rd Edition, 2008, ISBN: 9788131720011

General Guidelines

1. GOTO statements are not allowed
2. No global declarations allowed
3. Prototype for each user-defined-function must be provided before main
4. main should be the first function in any program
5. Students are encouraged to use user-defined-header files
6. Programs must be indented appropriately

Scheme of Continuous Internal Evaluation (CIE):

CIE consists of Three Tests each for 40 marks (15 marks for Quiz + 25 marks for descriptive) out of which best two will be considered. In addition 20 marks to be earned through self study component on emerging topics.

Scheme of Continuous Internal Evaluation for Practical's:

Cumulative Continuous Evaluation is done for a total of 50 marks. Out of that 30 marks are for execution of programs and record writing. Internal TEST Evaluation is for 20 marks.

Scheme of Semester End Examination (SEE):

The questions paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions one from each unit (descriptive, analytical, problems or/ and design) carrying 16 marks each. All five questions from Part B will have internal choice – one of the two questions has to be answered compulsorily.

Scheme of Semester End Examination for Practical's:

Students should select a program from the LOT of 14 programs. The student will be evaluated for 40 marks based on the Write-up, Execution, and Results. Viva Voce will be for 10 Marks.

Discrete Mathematical Structures

CourseCode : 12IS/CS 36 | **CIE Marks : 100**
L:T:P:S : 3:1:0:1 | **SEE Marks : 100**
Credits : 05 | **SEE Hours : 03**

Course Learning Objectives - CLO:

1. To provide an engineering student an intense foundational introduction to the fundamental concepts in Discrete Mathematics
2. To reinforce the close ties between discrete mathematics and the area of computer science.
3. To develop the mathematical maturity of the student through the study of an area that is different from the traditional coverage.
4. To present an adequate survey of topics for the Computer Science students who will be taking more advanced courses in areas such as data structures, the theory of computer languages, the analysis of algorithms, and fuzzy logic.

UNIT – I **08 Hrs**

Fundamental Principles of Counting: The Rules of Sum and Product, Permutations, Combinations, The Binomial Theorem, Combinations with Repetition.

Set Theory and the Principle of Inclusion and Exclusion: Sets and Subsets Set Operations and the Laws of Set Theory, Cartesian Products, The Principles of Inclusion and Exclusion, Generalizations of the Principle, Derangements – Nothing is in its Right Place, Rook Polynomials, and Arrangements with Forbidden Positions

UNIT – II **07 Hrs**

Mathematical Induction, Recursive Definitions and Recurrence Relations: Method of mathematical induction, Recursive definitions, First order Linear Recurrence Relation – Formulation Problems and examples. Second order Linear Homogeneous Recurrence Relations with Constant Coefficients. The Non-Homogeneous Recurrence Relations. The method of Generating Functions.

UNIT – III **07 Hrs**

Fundamentals of Logic: Basic Connectives and Truth Tables, Tautologies, Logical Equivalence: The laws of logic, Logical Implications: Rules of Inference, The use of Quantifiers. The proofs of Theorems

UNIT – IV **07 Hrs**

Relations: Properties of Relations, Composition of Relations, Partial Orders, Hasse Diagrams, Equivalence Relations and Partitions

Functions: Functions, Plain and One – to – one, Onto Functions: Stirling Numbers of the second kind, Function Composition and Inverse Function.

UNIT – V **07 Hrs**

Groups Theory: Definition, Examples, and Elementary Properties, Abelian Groups, Homomorphism, Isomorphism's, and Cyclic Groups, Cosets and Lagrange's Theorem.

Coding Theory: Elementary Coding Theory, The Hamming Metric, The Parity – Check and Generator Matrices, Group Codes: Decoding with Coset Leaders.

Course Outcome:

1. Students will demonstrate an ability of skills of discrete mathematics appropriate to the computer science discipline.
2. Students will demonstrate an understanding of emerging mathematical technologies and working knowledge of currently available mathematical tools like Languages, Finite state machines Coding theory.
3. Students will demonstrate the various techniques of finite state machines to compiler design.
4. Students will demonstrate the various techniques of finite state machines to compiler design.
5. The student is able to understand the role of discrete mathematical structures as a logical, predictable system for expressing andrelating quantities in analyzing and solving problems in the real world

Reference Books:

1. Ralph P. Grimaldi and B.V. Ramana, Discrete and Combinatorial Mathematics- An Applied Introduction, Pearson Education, Asia, Fifth Edition – 2007. ISBN 978-81-7758-424-0.
2. J.P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata – McGraw Hill, 35TH reprint 2008. ISBN 13:978-0-07-463113-3.
3. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata – McGraw Hill, Sixth Edition, Sixth reprint 2008. ISBN-(13):978-0-07-064824-1.
4. C.L. Liu and D.P.Mohapatra, Elementary Discrete Mathematics, Sixth Edition, Tata- McGraw Hill, ISBN:10:0-07-066913-9

Scheme of Continuous Internal Evaluation (CIE):

CIE consists of Three Tests each for 40 marks (15 marks for Quiz + 25 marks for descriptive) out of which best two will be considered. In addition 20 marks to be earned through self study component on emerging topics.

Scheme of Semester End Examination for Theory:

The question paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions carrying 16 marks each. All five questions from Part B will have internal choice and one of the two have to be answered compulsorily.

Bridge Course Mathematics- I
(All Branches)

Course Code: 12DMA37 **CIE Marks: 100**
Audit Course **SEE Marks: 100**
SEE : 3 Hrs

Course Learning Objectives:

- Apply the knowledge of ordinary and partial differentiation in engineering and real life problems;
- Learn how to formulate and interpret a Taylor series approximation of a function.
- Make the student to learn the concepts of vector analysis.
- Recognize and model differential equations, apply analytic techniques to compute solution for engineering problems.

UNIT – I **06 Hrs**

Differential Calculus: Successive differentiation, nth derivatives of standard functions, Leibnitz's theorem.Taylor's series and Maclaurin's series for function of single variable (all results without proof).

UNIT – II **06 Hrs**

Partial Differentiation: Introduction-partial derivatives, total derivative, differentiation of composite and implicit functions. Jacobiansand problems.

UNIT – III **08 Hrs**

Ordinary Differential Equations: Solution of first order and first degree differential equations - variable separable methods homogeneous, linear, Bernoulli, exact equations (without integrating factor).

UNIT – IV **06 Hrs**

Linear Ordinary Differential Equations of Second and Higher Order: Linear differential equations of higher order with constant coefficients. Solution by inverse differential operator method. Solution by method of variation of parameters.

UNIT – V **06 Hrs**

Vector Analysis: Vector Algebra - Vector addition, Multiplication (dot, cross & triple products), Vector differentiation – velocity, acceleration of a vector point function.

Course Outcomes:

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

- Use the concept of functions of several variables and their partial derivatives for computing the areas, volumes using multiple integrals.
- Ability to apply concept of differential equations to handle physical problems.

Reference Books:

1. B. S. Grewal, "Higher Engineering Mathematics", Khanna Publications, 40thEdition 2007.
2. N. P. Bali, Manish Goyal, "A Text Book of Engineering Mathematics", Laxmi Publications, 7th Edition, 2007.
3. B.V. Ramana, "Higher Engineering Mathematics", Tata McGraw Hill Publications, 2007.
4. E- Kreyszig "Advanced Engineering Mathematics", John Wiley &Sons Publications, 8thEdition, 2007.

Scheme of Continuous Internal Evaluation:
CIE consists of Two Tests each for 50 marks (20 marks for Quiz + 30 marks for descriptive).

Scheme of Semester End Examination:
The question paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions (descriptive, analytical, problems or/and design) carrying 16 marks each. All five from Part B will have internal choice and one of the two have to be answered compulsorily.

Semester: IV
Applied Mathematics IV
(EC/TC/IT/IS/CSE/ME/CHEMICAL)

Course Code: 12MA41 **CIE Marks: 100**
L:T:P:S: 3:1: 0:0 **SEE Marks: 100**
Credits: 04 **SEE : 3 Hrs**

Course Learning Objectives:

- Provide basic definitions and theorems of the calculus of complex functions which are involved in any field problems of Engineering.
- Use of Bessel functions and Legendre polynomials and their properties in Heat, wave and Laplace equations with cylindrical and spherical symmetry.
- The theory of probability in study of random phenomena, analyzing and interpreting data that involves uncertainities.
- Apply linear programming techniques for optimization problems subject to linear constraints in the various areas of Engineering & Science.
- A student will be able to find the solution of partial differential equations which arise in physical situations.

UNIT – I **07 Hrs**

Complex Analysis: Complex variables - Function of a complex variable, analytic functions-Cauchy-Riemann equations in cartesian and polar forms (without proof), properties of analytic functions, construction of analytic functions by Milne-Thomson method.
Complex integration - Complex line integrals-Cauchy's theorem and corollaries (without proof), Taylor's and Laurent's series (statements only), singularities, poles, residues, residue theorem (without proof) - problems.

UNIT – II **07 Hrs**

Special Functions: Introduction of Bessel's and Legendre's differential equation using the solution of Laplace equation in cylindrical and spherical system. Series solution of Bessel's differential equation leading to Bessel function of first kind, recurrence relations, generating functions, Bessel's integral formula, orthogonality of Bessel function. Legendre's differential equation, Legendre polynomials, Rodrigue's formula.

UNIT – III **07 Hrs**

Linear Programming Problem: Mathematical formulation of Linear Programming Problem, Graphical method, Simplex method and Big M method.

UNIT – IV **06 Hrs**

Probability and Distributions: Basics of Probability: Sample Space, events, probability of an event, addition theorem. Conditional probability, Multiplication theorem, Baye's rule. Random Variables: Discrete and continuous, Probability mass function, Probability density function, Cumulative density function, Mean, Variance, standard deviation Binomial, Poisson, Exponential and Normal Distributions.

UNIT – V **07 Hrs**

Partial Differential Equations: Classification of second order Partial differential equations - Elliptic, Parabolic and Hyperbolic. Solution of two dimensional Laplace equation in polar coordinates by the method of separation of variables. Solution of two dimensional heat flow in transient state and steady state. Solution of two dimensional wave equation by the method of separation of variables. Vibrating membrane, solution in the case of rectangular and circular membrane - Simple problems.

Course outcomes:

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

- Provide basic definitions and theorems of the calculus of complex functions which are involved in any field problems of Engineering.
- Use Bessel functions, Legendre polynomials and their properties in heat, wave and Laplace equations with cylindrical and spherical symmetry.
- Study of random phenomena, analyzing and interpreting data that involves uncertainty, using theory of probability.
- Interpret the models of probability distributions for real life and engineering problems.

Reference Books:

1. B.S. Grewal - Higher Engineering Mathematics, Khanna Publishers, 40th Edition, 2007, ISBN: 81-7409-195-5, Chapters: 16, 17, 19, 20, 26 and 33.
2. N.P Bali & Manish Goyal - A Text Book of Engineering Mathematics, Lakshmi Publications, 7thEdition, 2010, ISBN: 978-81-7008-992-6,Chapters: 15, 16, 21.
3. Erwin Kreyszig - Advanced Engineering Mathematics, John Wiley & Sons, 9th Edition, 2007, ISBN: 978-81-265-3135-6, Chapters: 4, 11, 12, 20 and 22.
4. Seymour Lipschutz& Marc Lars Lipson- Theory and Problems of Probability, Schaum's Outline Series, 2nd Edition, ISBN: 0-07—118356-6, Chapters: 1, 2, 3, 4,5and 6.

Scheme of Continuous Internal Evaluation:

CIE consists of Three Tests each for 45 marks (15 marks for Quiz + 30 marks for descriptive) out of which best of two will be considered. In addition there will be one seminar on new topics / model presentation etc. for 10 marks.

Scheme of Semester End Examination:

The question paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions (descriptive, analytical, problems or/and design) carrying 16 marks each. All five from Part B will have internal choice and one of the two have to be answered compulsorily.

Environmental Science and Biology for Engineers
(All Branches)

Course Code: 12EB42 **CIE Marks: 100**
L:P:T:S : 3:0:0:1 **SEE Marks: 100**
Credits: 04 **SEE : 3 Hrs**

Course Learning Objectives:

Suggested Objectives for Environmental Technology

1. To make engineering graduates understand the changes happening in the environment over decades (to give statistics with causes).
2. Role of human beings in the changes in environment and ways and means of controlling the changes through technology.
3. Sustainability issues in new technologies and its adaptation.
4. Innovation (case studies) to arrest degradation of environment

Suggested Objectives for Applied Biology:

1. To create awareness among all engineering graduates the need of biological study in engineering (biology related issues in each engineering profession with case studies and also application of biology in each program of engineering.
2. Various branches of biological sciences (this might contain discussion of basic human physiology, sensors and systems).
3. Effect of environment on biological issues and think of solutions (case studies in industrial environment to be studied)

UNIT – I **06 Hrs**

Ecosystems and Environment:Principles of ecosystem, impact of human being on environment: pollution, resource depletion and global environmental issues, ecosystem health and environmental changes and human health. Procedure to assess ecosystem's health. Standards- ISO14000 and Environmental Impact Assessment – definition, objectives, and types. Rapid and Comprehensive Environmental Impact Assessment (EIA), Environmental Impact Statement (EIS) and Finding Of No Significant Impact (FONSI). Some EIA examples –Thermal Power Plant, Mining, Fertilizer, Construction Projects, Airport, Water and Wastewater Treatment Plants.

UNIT – II **10 Hrs**

Strategies and Technology-based Solutions for Improvement of Environment Quality: Environment quality objectives and 'Waste challenge'in modern society - types of waste: municipal, agricultural, medicinal, E-waste, industrial. Engineering ethics, 3 R's – Reduce, Reuse & Recycle, and Sustainable waste management: Compacting, drying, dewatering, bio-drying, composting, bioremediation, biodegradation (chemicals and oil spillage). Waste to energy – energy recovery by incineration, bio-gasification, gasification and pyrolysis, bioconversion to clean energy (biofuels). Some examples: Upflow anaerobic sludge blanket (UASB) digestion for waste water treatment and biogas production.

Technology to Reduce Pollution: SO_2/CO_2 reduction by smoke-scrubber in coal thermal plants, chlorofluorocarbon (CFC) and incandescent bulb replacement, Renewable energy sources – wind, solar, tidal waves and biomass. Emerging technologies: Geo-engineering - ocean iron fertilization, green cement, bioremediation by terminator insects and synthetic biology.

UNIT – III **06 Hrs**

Design and Modeling for Development of Environment:Environmental Design: Principles, benefits and motivation. Environmental design for manufactured products, building and for developmental planning. Systems Engineering – Analysis - Design – synthesis - applications to environmental Engineering Systems.

EnvironmentalModeling: Introduction, forecast modeling and growth modeling, sensitivity analysis. Application of remote-sensing and geographic information systems (GIS) in environmental modeling.

UNIT – IV **06 Hrs**

Introduction to Cell and Organ Systems: Cell Types: Structure of plant, animal and microbial cell and specialized cells like stem cells and nerve cells. Biological macromolecules: Carbohydrates, proteins and nucleic acids and Special biomolecules – hormones, enzymes, vitamins and antibiotics. Introduction to organ systems for example digestive, respiratory, excretory nervous and circulatory. Nervous Control and coordination, sensory organs: Auditory, vision, olfactory, touch and taste.

UNIT – V **08 Hrs**

Bio-Inspired engineering (BIE) or Bionics:Biological phenomena and innovative engineering. Introduction to Bioelectronics, Biocomputing, biophotonics and biomechatronics. Locomotion and Bio-inspired Robotics, Prosthesis and biomedical implants, Aerodynamics and flight muscle functioning (birds & Drosophila).

Signaling: Enzymes and recognition receptors in biosensors; Neurotransmission and neural networks (artificial intelligence, signal processing and imaging); Bioelectric signals and cardiac generator.

Sound: Ultrasonics in biology (echolocation in bats, sonar in whales & dolphins) and instrumentation (medical ultrasonography - ultrasound imaging).

Light: Photosynthesis and photovoltaic cells

Course outcomes:

The graduates will have the knowledge of:

1. The adverse changes in the environment due to human activities.
2. The need of innovative technology to arrest or reverse these changes.
3. Ethical considerations important for systems engineering.
4. Basics of biological phenomena.
5. Their application in innovative engineering and development of technology.

Reference Books:

1. Vijay Kulkarni and T. V. Ramachandra 2009. Environment Management. TERI Press; ISBN: 8179931846, 9788179931844.
2. Gerald Kiely 1997. Environmental Engineering. McGraw-Hill; ISBN: 9780077091279.
3. Sven Erik Jørgensen 2002. Integration of Ecosystem Theories: A Pattern Ecology & Environment; Edition 3, Springer; ISBN: 1402007558, 9781402007552.
4. Linvil Gene Rich 2003. *Environmental Systems Engineering*, McGraw-Hill; ISBN: 9780070522503.
5. Ni-Bin Chang: Systems Analysis for Sustainable Engineering: Theory and Applications (Green Manufacturing & Systems Engineering). McGraw-Hill Professional.
6. Larry Canter 1995. “Environmental Impact Assessment”

Scheme of Continuous Internal Evaluation:

CIE consists of Three Tests each for 45 marks (15 marks for Quiz + 30 marks for descriptive) out of which best of two will be considered. In addition there will be one seminar on new topics / model presentation etc. for 10 marks.

Scheme of Semester End Examination:

The question paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions (descriptive, analytical, problems or/and design) carrying 16 marks each. All five from Part B will have internal choice and one of the two have to be answered compulsorily.

Theory of Computation

Course Code : 12IS/CS 43 **CIE Marks : 100**
L:T:P:S : 3:0:0:1 **SEE Marks : 100**
Credits : 04 **SEE Duration: 03**

Course Learning Objectives (CLO):

1. Understand fundamental concepts of theory of computation and the use of mathematical thinking as it is applied to Computer Science.
2. Compare finite automata; push down automata and Turing machines as Mathematical models of computation.
3. Develop the concepts and skills necessary to be able to evaluate the computability and complexity of practical computational problems.
4. Understand formal thought processes, computation, algorithms and their limits.
5. Design a machine model to accept a specified language

UNIT – I **08 Hrs**

Introduction to Finite Automata, Basic concepts of Automata theory, Deterministic Finite Automata, Non-Deterministic Finite Automata, Finite Automata with epsilon-transitions, Equivalence of NFA & DFA. Equivalence & minimization of automata.

UNIT – II

07Hrs Regular Expressions, Finite Automata & Regular Expressions, Applications of Regular Expressions, Algebraic laws of RE, Pumping Lemma, Closure properties of Regular Languages, Decision properties of Regular languages.

UNIT – III **08 Hrs**

Context-free grammars, Parse trees, Applications, Ambiguity in grammars &languages,Normal forms of CFGs, The Pumping Lemma, Closure properties of CFLs, Decision properties of CFLs.

UNIT – IV **06 Hrs**

Definition, the languages of a PDA, Equivalence of PDA's & CFG's, Deterministic PDA.

UNIT- V **07Hrs**

Problems that computers cannot solve, The Turing Machine and basic concepts, Programming techniques for Turing Machines, Different types of Turing Machines, Turing Machines & Computers. A language that is not recursively enumerable, An undecidable problem that is RE, Post's Correspondence problem, Other undecidable problems. The Chomsky hierarchy.

Course Outcome:

1. Student will be able toapply a number of proof techniques to theorems in language design.
2. Develop a clear understanding of undecidability.
3. Understand the equivalence between Non-deterministic Finite State Automata and Deterministic Finite State Automata.
4. Understand the equivalence between Context-Free Grammars and Non-deterministic Pushdown Automata.
5. Appreciate the power of the Turing Machine, as an abstract automaton, that describes computation, effectively and efficiently

Reference Books:

1. J.P.Hopcroft, Rajeev Motwani, J.D.Ullman, Introduction to Automata Theory, Languages& Computation, 3rdEdition, Pearson Education.

2. John Martin, Introduction to Languages & Theory of Computation, Tata McGraw-Hill, 3rdEdition, 2007.
3. Peter Linz, An Introduction to Formal Languages & Automata, VI Edition, Narosa Publishing House, 2007.

Scheme of Continuous Internal Evaluation:

CIE consists of Three Tests each for 40 marks (15 marks for Quiz + 25 marks for descriptive) out of which best two will be considered. In addition 20 marks to be earned through self learning component on emerging topics.

Scheme of Semester End Examination:

The question paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions (descriptive, analytical, problems or/and design) carrying 16 marks each. All five from Part B will have internal choice and one of the two have to be answered compulsorily.

Computer Organization and Architecture

Subject Code : 12IS44 | **CIE Marks : 100**
L:T:P:S : 3:0:0:0 | **SEE Marks : 100**
Credits : 03 | **SEE Hours : 03**

Course Learning Objectives-CLO:

1. Understand the main components of computers and the basic principles of their operation and Interconnection Structures that realize the architecture.
2. Analyze the relationship between hardware design and instruction set architecture.
3. Explore and apply the methods for evaluating and comparing processor performance.
4. Provide a comprehensive coverage of Parallel Processing and Multi-core Architecture.

UNIT – I **07 Hrs**

A Top-Level view of Computer Function and Interconnection: Computer Components, Computer Function, Interconnection structures, Bus Interconnection, PCI.
Cache Memory: Computer Memory System Overview, Cache Memory Principles, Elements of Cache Design, Pentium 4 and PowerPC Cache Organization.

UNIT – II **08 Hrs**

Internal Memory: Semiconductor Main Memory, Error Correction, Advanced DRAM Organization.
External Memory: Magnetic Disk, RAID, Optical Memory, Magnetic Tape.
Input/Output: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels and Processors, External Interface: FireWire and InfiniBand.

UNIT – III **07 Hrs**

Computer Arithmetic: The Arithmetic and Logic Unit, Integer Representation, Integer Arithmetic, Floating-Point Representation, Floating-Point Arithmetic.
Processor Structure and Function: Processor Organization, Register Organization, Instruction Cycle, Instruction Pipelining.

UNIT – IV **07 Hrs**

Control Unit Operation: Micro-Operations, Control of the Processor, Hardwired Implementation.
Microprogrammed Control: Basic Concepts, Microinstruction Sequencing, Microinstruction Execution.

UNIT – V **06 Hrs**

Parallel Processing and Performance Considerations: Multiple Processor Organizations, Symmetric Multiprocessors, Cache Coherence and the MESI Protocol, Multithreading and Chip Multiprocessors, Clusters, Non-uniform Memory Access, Vector Computation, Performance Consideration: Amdahl's Law, Performance Indicators.

Software Development Products: Introduction, Debugging, Tuning, Thread Checker, Compilers, Libraries, Optimization techniques, performance Analyzer and Load balancing, Thread profiler, MPI programming.

Course Outcomes:

1. Demonstrate the relationship between software and hardware and focuses on the fundamental concepts that are the basis for current computer design.
2. Describe various data representation and explain how arithmetic and logical operations are performed by computers.
3. Evaluate the performance of CPU, memory and I/O operations.
4. Understand parallelism both in terms of a single processor and multiple processors.

Reference Books:

1. William Stallings, "Computer Organization and Architecture", 7th Edition, PHI, 2012, ISBN: 9788131732458.
2. Shameem Akhter and Jason Roberts, "Multi-core Programming", Intel Press, 2006, ISBN: 0-9764832-4-6.
3. Carl Hamacher, Z Vranesic& S Zaky, "Computer Organization", 5th edition, McGraw Hill, 2012, ISBN: 9781259005275.
4. David A. Patterson and John L. Hennessy, "Computer Organization and Design", 4th Edition, Elsevier, 2012, ISBN: 9780123747501.

Scheme of Continuous Internal Evaluation (CIE):

CIE consists of Three Tests each for 45 marks (15 marks for Quiz + 30 marks for descriptive) out of which best two will be considered. In addition, there will be one Assignment/Seminar on new topics/nodal presentation etc. for 10 marks.

Scheme of Semester End Examination (SEE):

The question paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions (descriptive, analytical, problems or/and design) carrying 16 marks each. All five questions from Part B will have internal choice and one of the two have to be answered compulsorily.

Design and Analysis of Algorithms
(Theory and Practice)

Course Code : 12IS/CS 45	**CIE Marks : 100 + 50**
L:T:P:S : 3:0:1:1	**SEE Marks : 100 + 50**
Credits : 05	**SEE Hours : 03+03**

Course Learning Objectives - CLO:

1. To learn a mathematical model to find complexity of algorithms
2. To learn different algorithm design techniques and the algorithms that employs these techniques.
3. To analysis the efficiency of the algorithms using the mathematical model developed.
4. To understand the limitations of algorithmic power

Prerequisites:

A strong understanding of programming and a solid background in discrete mathematics, including probability, are necessary prerequisites to this course.

UNIT – I **08 Hrs**

Fundamentals of Algorithm Analysis: Definition of algorithm, Algorithmic Problem Solving, Framework for Analysis of algorithm efficiency.

Asymptotic Notations : O, Ω, and θ notations, Properties of asymptotic notations, Basic Efficiency Classes, Mathematical Analysis of Non recursive algorithms, Mathematical Analysis of Recursive Algorithms and Introduction to Divide and Conquer, Master Theorem, Empirical analysis of algorithms.

UNIT – II **07 Hrs**

Mergesort, Quicksort, Comparison with brute force selection sort, Multiplication of large integers,

Decrease and Conquer : Depth First Search(DFS), Breadth First Search(BFS), Applications of DFS and BFS, Topological Sorting, Fake coin Problem, Computing a median and selection problem,

Transform and Conquer : Introduction, Presorting, Balanced Search Trees

UNIT – III **07 Hrs**

Heaps and Heapsort, Problem reduction

String Matching – Brute force method, Horspool's method, Boyer-Moore's method

DynamicProgramming(DP):Computing a Binomial Coefficient, Floyd's Algorithm, The Knapsack Problem – Brute force method, bottom-up DP method and Memory Functions.

UNIT – IV **07 Hrs**

Greedy Technique: Introduction, Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees**.**

Iterative improvement – Introduction, The maximum-flow problem, maximum matching in bipartite graphs, Lower-Bound Arguments, Decision Trees

UNIT – V **07 Hrs**

P, NP and NP-Complete Problems.

Backtracking(BT): n-queens problem, Subset-Sum problem,

Branch-and-Bound(BB): Assignment problem - Knapsack problem, Traveling Salesman Problem.

Laboratory Component

Implement the following using C++ Language.

1. Write a program to sort a given set of elements using Merge sort method and find the time required to sort the elements.

2. Write a program to print all the nodes reachable from a given starting node in a graph using Depth First Search method. Also check connectivity of the graph. If the graph is not connected, display the number of components in the graph.

3. Write a program to obtain the Topological ordering of vertices in a given digraph using
a) Vertices deletion method
b) DFS method

4. Write a program to sort a given set of elements using Heap sort method. Find the time complexity.

5. a) Write a program to implement Horspool algorithm for String Matching.
b) Write a program to implement all pair shortest paths problem using Floyd's algorithm.

6. Write a program to implement 0/1 Knapsack problem using dynamic programming.

7. Write a program to find Minimum cost spanning tree of a given undirected graph using Kruskal's algorithm.

8. Write a program to find the shortest path using Dijkstra's algorithm for a weighted connected graph.

9. Write a program to implement Subset-Sum problem using Back Tracking.

10. Write a program to implement TSP using branch and bound algorithm.

11. Write a program to print all the nodes reachable from a given starting node in a graph using Breadth First Search method. Also check connectivity of the graph. If the graph is not connected, display the number of components in the graph.

12. Write a program to find Minimum cost spanning tree of a given undirected graph using Prim's algorithm.

13. Write a program to implement n-queens problem

14. Write a program to sort a given set of elements using Quick sort method and find the time required to sort the elements

Course Outcome:

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

1. Do the complexity analysis of algorithms.
2. Identify the efficient design technique to solve a given problem.
3. Understand the limitation of the algorithms and techniques to overcome it.

Reference Books:

1. Levitin A., Introduction to The Design and Analysis of Algorithms, Pearson Education,

2003,ISBN: 9780201743951
2. Cormen T.H., Leiserson C. E., Rivest R.L., Stein C., Introduction to Algorithms, 3rd edition, PHI 2010, ISBN:9780262033848
3. Horowitz E., Sahani S., Rajasekharan S., Computer Algorithms,Galgotia Publications, 2001,ISBN: 9780716783169
4. Mark Allen Weiss, Data structures and algorithm analysis in C++, Pearson Education, 2003,ISBN: 032144146

General Guidelines

1. GOTO statements are not allowed
2. No global declarations allowed
3. Prototype for each user-defined-function must be provided before main
4. Main should be the first function in any program
5. Students are encouraged to use user-defined-header files
6. Programs must be indented appropriately
7. Students are required to bring only the algorithms in the data sheet

Scheme of Continuous Internal Evaluation (CIE):

CIE consists of Three Tests each for 40 marks (15 marks for Quiz + 25 marks for descriptive) out of which best two will be considered. In addition 20 marks to be earned through self study component on emerging topics.

Scheme of Continuous Internal Evaluation for Practical's:

Cumulative Continuous Evaluation is done for a total of 50 marks. Out of that 30 marks are for execution of programs and record writing. Internal TEST Evaluation is for 20 marks.

Scheme of Semester End Examination (SEE):

The questions paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions one from each unit (descriptive, analytical, problems or/ and design) carrying 16 marks each. All five questions from Part B will have internal choice – one of the two questions has to be answered compulsorily.

Scheme of Semester End Examination for Practical's:

Students should select a program from the LOT of 14 programs. The student will be evaluated for 40 marks based on the Write-up, Execution, and Results. Viva Voce will be for 10 Marks.

Operating Systems

Course Code:12IS46 **CIE Marks :100**
L:T:P:S : 3:0:0:0 **SEE Marks :100**
Credits:03 **SEE Duration:03Hrs**

Course Learning Objectives (CLO):

1. To introduce Operating System concepts with emphasis on foundations & design principles.
2. To compare, contrast, and evaluate the key trade-offs between multiple approaches to operating system design, and identify appropriate design choices when solving real-world problems.
3. To motivate and explain efficient programming concepts, relevant to process, memory and disk management.
4. To analyze various file system interfaces and implementation, disk management, protection and security mechanisms and apply to solve real-world problem

UNIT – I **07 Hrs**

Introduction to Operating Systems, System Structures:
Functions of an operating system; Computer System organization; Computer System architecture; Operating System structure and operations; Process management; Memory management; Storage management; Protection and security; Distributed system; Special purpose systems; Computing environments. Operating System Services; User - Operating System interface; System calls; Types of system calls; System programs; Operating System design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot.

UNIT – II **07 Hrs**

Process Management: Process concept, Scheduling, Operations; Inter-process communication. Multi-Threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling criteria; Scheduling algorithms; Multiple-Processor scheduling; thread scheduling.

UNIT – III **07 Hrs**

Process Synchronization: Synchronization: The Critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors.

UNIT – IV **08 Hrs**

Deadlocks: Deadlocks: System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock.
Memory Management: Memory Management Strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation. Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing.

UNIT – V **07 Hrs**

File System, Implementation of File System: File System: File concept; Access methods; Directory structure; File system mounting; File sharing; Protection. Implementing File System: File system structure; File system implementation; Directory implementation; Allocation methods; Free space management.
Secondary Storage Structures, Protection: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability-Based systems.

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

1. Compare and contrast the various ways of structuring an operating system such as modular, micro-kernel, and layered, and the benefits of building abstract layers in hierarchical fashion
2. Summarize techniques for achieving synchronization in an operation system and compare common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems, such a priority, performance comparison, and fair-share schemes.
3. Design programs which include concepts like multiple concurrent threads and/or processes, file operations and analyze how it is realized in hardware and software, including caching, paging, and segmentation.
4. Design programs using different ways of allocating memory to tasks, including overlays, swapping, and placement and replacement policies, citing the relative merits of each.

Reference Books:

1. Abraham Silberschatz, Peter Baer Galvin,Greg Gagne: Operating System Principles, 8th Edition, Wiley-India, 2011, ISBN:1118112733, 9781118112731.
2. Tanenbaum, Woodhull: Operating Systems-Design and Implementation, 3rd Edition, Prentice Hall Edition,2006. ISBN 10: 0131429388 / ISBN 13: 9780131429383.
3. Harvey M. Deitel, Paul J. Deitel, David R. Choffnes:Operating Systems, 3rd Edition, Pearson Education Pvt. Ltd,2007.ISBN :9788131712894.

Scheme of Continuous Internal Evaluation(CIE):Theory

CIE consists of Three Tests each for 45 marks(15 marks for Quiz+30 marks for descriptive) out of which best of two will be considered. In addition,there will be one Assignment/Seminar on new topics/model presentation etc. for 10 marks.

Scheme of Semester End Examination (SEE):Theory

The question paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five question(descriptive,analytical, problems or/and design) carrying 16 marks each. All five from Part B will have internal choice and one of the two have to be answered compulsorily.

Bridge Course Mathematics- II
(All Branches)

Course Code: 12DMA48 **CIE Marks: 100**
Audit Course **SEE Marks: 100**
SEE : 3 Hrs

Course Learning Objectives:

- Recognize partial differential equations and apply analytic techniques to compute solution for engineering problems.
- Apply the significance of vector differentiation and their theoretical importance in engineering problems.
- Apply the significance of Laplace transforms and inverse Laplace transforms and their theoretical importance in engineering problems.
- Identify and solve initial value problems, physically interpret the solutions using theLaplace transforms.

UNIT – I **06 Hrs**

Laplace Transforms: Definition, transforms of elementary functions, properties, derivatives and integrals,unit step function.

UNIT – II **06 Hrs**

Inverse Laplace Transforms: Inverse Laplace transforms- properties, convolution theorem (statement only) - problems. Solution of linear differential equations with constant coefficients.

UNIT – III **06 Hrs**

Integral Calculus: Multiple integrals - Double and Triple integrals. Area enclosed by plane curves, Volume of solids. Definition of beta and gamma functions and problems.

UNIT – IV **06 Hrs**

Partial Differential Equations (PDE): Formation of Partial differential equations by elimination of arbitrary constants/functions. Solution of Lagrange's linear PDE. Solution of PDE by the Method of separation of variables (first and second order equations).

UNIT – V **06 Hrs**

Vector Analysis: Vector Differentiation - Scalar and vector point functions, gradient, directional derivative, divergence and curl. Solenoidal and Irrotational fields, Vector identities.

Course Outcomes:

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

- The student will be able to solve problems arising in signal processing and various systems using Laplace transforms techniques for problems arising in signals and systems.
- The student will be able to apply vector integration to different Engineering applications.

Reference Books:

1. B. S. GREWAL, "Higher Engineering Mathematics", Khanna Publications, 40thEdition 2007.
2. N. P. BALI, MANISH GOYAL "A Text Book of Engineering Mathematics", LaxmiPublications, 7th Edition, 2007.
3. B. V. RAMANA "Higher Engineering Mathematics", Tata McGraw Hill Publications, 2007.
4. E- KREYSZIG "Advanced Engineering Mathematics", John Wiley & Sons Publications, 8th Edition, 2007.

Scheme of Continuous Internal Evaluation:

CIE consists of Two Tests each for 50 marks (20 marks for Quiz + 30 marks for descriptive).

Scheme of Semester End Examination:

The question paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions (descriptive, analytical, problems or/and design) carrying 16 marks each. All five from Part B will have internal choice and one of the two have to be answered compulsorily.

UNIX Systems Programming

(Theory and Practice)

Course Code : 12IS49 **CIE Marks : 100 + 50**

L:T:P:S : 3:0:1:1 **SEE Marks : 100 + 50**

Credits : 05 **SEE Hours : 03+03**

Course Learning Objectives (CLO):

1. Students study the architecture of UNIX Operating System and its use.
2. Apply the different file handling APIs for different applications.
3. Students study the concepts of IPC, Process management, Semaphores and implement the same.
4. Analyze Client Server communication and its applications.

PART –A (Theory)

UNIT – I **07 Hrs**

UNIX and ANSI Standards: The ANSI C Standard, the ANSI/ISO C++ Standards, Difference between ANSI C and C++, the POSIX Standards, the POSIX.1 FIPS Standard, The X/Open Standards.

UNIX and POSIX APIs: The POSIX APIs, The UNIX and POSIX Development Environment, API Common Characteristics.

UNIX Files: File Types, The UNIX and POSIX File System, The UNIX and POSIX File Attributes, Inodes in UNIX System V, Application Program Interface to Files, UNIX Kernel Support for Files, Relationship of C Stream Pointers and File Descriptors, Directory Files, Hard and Symbolic Links.

UNIT – II **07 Hrs**

UNIX File APIs: General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs, General File Class, regfile Class for Regular Files, dirfile Class for Directory Files, FIFO File Class, Device File Class, Symbolic Link File Class, File Listing Program.

UNIX Processes:

The Environment of a UNIX Process: Introduction, main function, Process Termination, Command-Line Arguments, Environment List, Memory Layout of a C Program, Shared Libraries, Memory Allocation, Environment Variables, setjmp and longjmp Functions, getrlimit, setrlimit Functions, UNIX Kernel Support for Processes.

UNIT – III **07 Hrs**

Process Control: Introduction, Process Identifiers, fork, vfork, exit, wait, waitpid, wait3, wait4 Functions, RaceConditions, exec Functions, Changing User IDs and Group IDs, Interpreter Files, system Function, Process Accounting, User Identification, Process Times, I/O Redirection.

Process Relationships: Introduction, Terminal Logins, Network Logins, Process Groups, Sessions, Controlling Terminal, tcgetpgrp and tcsetpgrp Functions, Job Control, Shell Execution of Programs, Orphaned Process Groups.

Signals: The UNIX Kernel Support for Signals, signal, Signal Mask, sigaction, The SIGCHLD Signal and the waitpid Function, The sigsetjmp and siglongjmp Functions, Kill, Alarm, Interval Timers, POSIX.lb Timers.

UNIT – IV 07 Hrs

Daemon Processes: Introduction, Daemon Characteristics, Coding Rules, Error Logging, Client-Server Model.

Inter-process Communication: Overview of IPC Methods, Pipes, popen, pclose Functions, Coprocesses, FIFOs, System V IPC, Message Queues, Semaphores, Shared Memory, Client-Server Properties, Stream Pipes, Passing File Descriptors, An Open Server-Version 1, Client-Server Connection Functions.

UNIT – V 08 Hrs

Remote Procedure Calls: History of RPC, RPC Programming Interface Levels, RPC Library Functions, rpcgen – clnt_create, the rpcgen Program, a directory Listing Example Using rpcgen, rpcgen limitations, Lower Level RPC Programming Interfaces – XDR Conversion Functions, Lower Level RPC API's.

Multithreaded Programming: Thread structure and Uses, Threads and Lightweight Processes, POSIX.1c Thread APIs–thread_create, pthread_exit, pthread_detach, pthred_join, pthread_sigmask, pthread_kill, sched_yield.

Thread Synchronization Objects: Mutually Exclusive Locks (mutex Locks), POSIX.1c Mutex Locks, Mutex Lock Examples, POSIX.1c Conditional Variables.

Laboratory Component

1 a. Write a C program that creates a child process using fork () system call and print different messages from child and parent process. The child process should read commands from the standard input and execute them (a minimal implementation of a shell – like program).
   b. Write a C program that accepts valid file names as command line arguments and for each of the arguments, prints the type of the file (Regular file, directory file, Character special file, Block special file, symbolic link etc.,)

2 a. Write a C program to do the following: using fork () create a child process. The child process prints its own process-id and id of its parent and then exits. The parent process waits for its child to finish (by executing the wait ()) and prints its own process-id and the id of its child process and then exists.
   b. Implement a program to execute a program periodically by a daemon.

3 a. Write a program to examine and modify process priorities with getpriority(), setpriority(). Demonstrate the impact of change in priority on the output.
   b. Write a program to demonstrate two way Inter Process Communications between two processes using FIFO.

4 a. Implement file transfer using Message Queue form of IPC. The client process request for a file that is sent by the Server process. Server returns a suitable message if the file is not available.
   b. Write a program which illustrates the signal handling for SIGFPE signal.

5 a. Examining and modifying process priorities and scheduling policy with sched_gettscheduler(), sched_setscheduler().
   b. Write a program to demonstrate two way Inter Process Communications between two processes using pipes. Use SELECT system call to monitor the read-ends of the pipes

6 a. Write a program to demonstrate Use semaphores to avoid race conditions. (Hint: Create an integer variable in Shared Memory that is incremented by multiple processes causing race condition)
   b. Implement a counting semaphore using the binary semaphores.

7 Write a C program to solve Dining Philosophers Problem.

8 Write a C program to solve Producer – Consumer problem using semaphores

9 Write C programs to implement the CPU scheduling algorithms for FCFS Scheduling and SJF Scheduling. Compare their performance metrics in terms of average turnaround time, average waiting time and average weighted turnaround.

10 Write C programs to implement the CPU scheduling algorithms for priority Scheduling and Round robin Scheduling. Compare their performance metrics in terms of average turnaround time, average waiting time and average weighted turnaround.

11 Write a C program to implement the Banker's algorithm for deadlock-avoidance and verify whether a system is in a safe state.

12 Write C programs to implement the FIFO and Optimal Page replacement algorithms and compare them based on the number of page faults for sample reference strings.

13 Write C programs to implement the FIFO and SCAN disk scheduling algorithms. Compare the disk scheduling algorithms based on the total head movement for given cylinder numbers

14 Implement readers-writers problem using semaphores. Use counting semaphore to allow more than one reader read.

Course Outcome:

1. Students will be familiar with UNIX operating system and its features.
2. Students study advanced UNIX system programming and be able to solve system related problems.
3. Students will be able to code client server applications on different network protocols.
4. Analyze system and program performance and optimize the code.

Reference Books:

1. Terrence Chan: UNIX System Programming Using C++, Prentice Hall India, 2008.
2. W.Richard Stevens: Advanced Programming in the UNIX Environment, Addison-Wesley, 2nd Edition, 2007.
3. Sumitabha Das, UNIX Concepts and Applications, 4th Edition, Tata McGraw Hill, 2010.
4. Maurice.J.Bach: The Design of the UNIX Operating System, Prentice Hall of India, 3rdEdition.

General Guidelines

1. GOTO statements are not allowed
2. No global declarations allowed
3. Prototype for each user-defined-function must be provided before main
4. main should be the first function in any program
5. Students are encouraged to use user-defined-header files
6. Programs must be indented appropriately

Scheme of Continuous Internal Evaluation (CIE):

CIE consists of Three Tests each for 40 marks (15 marks for Quiz + 25 marks for descriptive) out of which best two will be considered. In addition 20 marks to be earned through self study component on emerging topics.

Scheme of Continuous Internal Evaluation for Practical's:

Cumulative Continuous Evaluation is done for a total of 50 marks. Out of that 30 marks are for execution of programs and record writing. Internal TEST Evaluation is for 20 marks.

Scheme of Semester End Examination (SEE):

The questions paper consists of Part A and Part B. Part A will be for 20 marks covering the complete syllabus and is compulsory. Part B will be for 80 marks and shall consist of five questions one from each unit (descriptive, analytical, problems or/ and design) carrying 16 marks each. All five questions from Part B will have internal choice – one of the two questions has to be answered compulsorily.

Scheme of Semester End Examination for Practical's:

Students should select a program from the LOT of 14 programs. The student will be evaluated for 40 marks based on the Write-up, Execution, and Results. Viva Voce will be for 10 Marks.