

SYLLABUS
II Year- III Semester : Information Technology

3IT2-01 : Advanced Engineering Mathematics

Credit- 3
3L+0T+0P

Max. Marks : 150 (IA:30, ETE:120)
End Term Exam: 03 Hours

COURSE ASSESSMENT METHOD: Two midterm and one end term examination	
PRE-REQUISITES: Basic understanding of Probability and calculus	
COURSE OBJECTIVES:	
1.	To learn the concepts and principles of Random variable and Probability distribution
2.	To learn the formulation of different mathematical problems into optimization Problems
3.	Apply the principles of optimization using differential calculus
4.	To understand the concepts of Liner Programming

SN	CONTENTS	Hours
1	Random Variables: Discrete and Continuous random variables, Joint distribution, Probability distribution function, conditional distribution. Mathematical Expectations: Moments, Moment Generating Functions, variance and correlation coefficients, Chebyshev's Inequality, Skewness and Kurtosis.	7
2	Binomial distribution , Normal Distribution, Poisson Distribution and their relations, Uniform Distribution, Exponential Distribution. Correlation: Karl Pearson's coefficient, Rank correlation. Curve fitting. Line of Regression.	5
3	Historical development , Engineering Applications of Optimization, Formulation of Design Problems as a Mathematical Programming Problems, Classification of Optimization Problems	8

4	Classical Optimization using Differential Calculus: Single Variable and Multivariable Optimization with & without Constraints, Langrangian theory, Kuhn Tucker conditions	6
5	Linear Programming: Simplex method, Two Phase Method and Duality in Linear Programming. Application of Linear Programming: Transportation and Assignment Problems.	14
TOTAL		40

3IT1-02/4IT1-02: Technical Communication

Credit- 2
2L+0T+0P

Max. Marks : 100 (IA:20,ETE:80)
End Term Exam: 2 Hours

COURSE ASSESSMENT METHOD: Two midterm and one end term examination

PRE-REQUISITES:

SN	CONTENTS	Hours
1	Vocabulary Building: Concept of Word Formation, Affixes, Synonyms and Antonyms.	5
2	Grammar Words and Sentences, Verbs and Tenses, Questions and Question Tags, The Infinitive and the ‘...ing’ form.	5
3	Grammar Nouns and Articles, Determiners, Adjectives and Adverbs, Relative clauses.	6
4	Identifying Common Errors in Writing Subject-Verb, Agreement, Noun-Pronoun Agreement, Articles, Prepositions.	5
5	Composition Précis Writing, Essay Writing, Comprehension of Passage.	5
TOTAL		26

3IT1-03/ 4IT1-03 : Managerial Economics and Financial Accounting

Credit- 2
2L+0T+0P

Max. Marks : 100 (IA:20, ETE:80)
End Term Exam: 2 Hours

COURSE ASSESSMENT METHOD: Two midterm and one end term examination	
PRE-REQUISITES: Basic Understanding of economics	
COURSE OBJECTIVES:	
1.	To understand the basic concepts of economics
2.	To understand the relation between demand and supply
3.	To learn the concepts of production and cost analysis
4.	To understand financial statement analysis

SN	CONTENTS	Hours
1	Basic economic concepts- Meaning, nature and scope of economics, deductive vs inductive methods, static and dynamics, Economic problems: scarcity and choice, circular flow of economic activity, national income-concepts and measurement.	5
2	Demand and Supply analysis- Demand-types of demand, determinants of demand, demand function, elasticity of demand, demand forecasting –purpose, determinants and methods, Supply-determinants of supply, supply function, elasticity of supply.	5
3	Production and Cost analysis- Theory of production- production function, law of variable proportions, laws of returns to scale, production optimization, least cost combination of inputs, isoquants. Cost concepts-explicit and implicit cost, fixed and variable cost, opportunity cost, sunk costs, cost function, cost curves, cost and output decisions, cost estimation.	6

4	Market structure and pricing theory- Perfect competition, Monopoly, Monopolistic competition, Oligopoly.	5
5	Financial statement analysis- Balance sheet and related concepts, profit and loss statement and related concepts, financial ratio analysis, cash-flow analysis, funds-flow analysis, comparative financial statement, analysis and interpretation of financial statements, capital budgeting techniques.	5
TOTAL		26

3IT3-04 : Digital Electronics

Credit- 3
3L+0T+0P

Max. Marks : 150 (IA:30, ETE:120)
End Term Exam: 3 Hours

COURSE ASSESSMENT METHOD: Two midterm and one end term examination	
PRE-REQUISITES: Basic understanding of Electrics and Electronics.	
COURSE OBJECTIVES:	
1.	To acquaint the students with the fundamental principles of two-valued logic and various devices used to implement logical operations on variables.
2.	Acquire knowledge about various logic gates and logic families and analyze basic circuits of these families.
3.	Develop ability to identify, analyze and design combinational circuits.
4.	Develop ability to design various synchronous and asynchronous sequential circuits.
5.	To prepare students to perform the analysis and design of various digital electronic Circuits.

SN	CONTENTS	Hours
1	<p>Fundamental concepts:</p> <p>Number systems and codes, Basic logic Gates and Boolean algebra: Sign & magnitude representation, Fixed point representation, complement notation, various codes & arithmetic in different codes & their inter conversion. Features of logic algebra, postulates of Boolean algebra. Theorems of Boolean algebra.</p>	8
2	<p>Minimization Techniques and Logic Gates:</p> <p>Principle of Duality - Boolean expression -Minimization of Boolean expressions — Minterm – Maxterm - Sum of Products (SOP) – Product of Sums (POS) – Karnaugh map Minimization – Don't care conditions – Quine - McCluskey method of minimization.</p>	8
3	<p>Digital Logic Gate Characteristics:</p> <p>TTL logic gate characteristics. Theory & operation of TTL NAND gate circuitry. Open collector TTL. Three state output logic. TTL subfamilies. MOS& CMOS logic families. Realization of logic gates in</p>	8

	RTL, DTL, ECL, C-MOS & MOSFET.	
4	Combinational Circuits: Combinational logic circuit design, adder, subtractor, BCD adder encoder, decoder, BCD to 7-segment decoder, multiplexer demultiplexer.	8
5	Sequential Circuits: Latches, Flip-flops - SR, JK, D, T, and Master-Slave Characteristic table and equation, counters and their design, Synchronous counters – Synchronous Up/Down counters – Programmable counters – State table and state transition diagram ,sequential circuits design methodology. Registers –shift registers.	8
TOTAL		40

3IT4-05 : Data Structures and Algorithms

Credit- 3
3L+0T+0P

Max. Marks : 150 (IA:30, ETE:120)
End Term Exam: 3 Hours

COURSE ASSESSMENT METHOD: Two midterm and one end term examination	
PRE-REQUISITES: Basic understanding of C- Programming	
COURSE OBJECTIVES:	
1.	To impart the basic concepts of data structures and algorithms
2.	To understand concepts about searching and sorting techniques
3.	To Understand basic concepts about stacks, queues, lists, trees and graphs
4.	To understanding about writing algorithms and step by step approach in solving problems with the help of fundamental data structures
5.	To impart the basic concepts of data structures and algorithms

SN	CONTENTS	Hours
1	Stacks: Basic Stack Operations, Representation of a Stack using Static Array and Dynamic Array, Multiple stack implementation using single array, Stack Applications: Reversing list, Factorial Calculation, Infix to postfix Transformation, Evaluating Arithmetic Expressions and Towers of Hanoi.	8
2	Queues: Basic Queue Operations, Representation of a Queue using array, Implementation of Queue Operations using Stack, Applications of Queues- Round Robin Algorithm. Circular Queues, DeQueue Priority Queues. Linked Lists: Introduction, single linked list, representation of a linked list in memory, Different Operations on a Single linked list, Reversing a single linked list, Advantages and disadvantages of single linked list circular linked list, double linked list and Header linked list.	10
3	Searching Techniques: Sequential and binary search. Sorting Techniques: Basic concepts, Sorting by: bubble sort, Insertion sort	7

	selection sort, quick sort, heap sort, merge sort, radix sort and counting sorting algorithms.	
4	Trees: Definition of tree, Properties of tree, Binary Tree, Representation of Binary trees using arrays and linked lists, Operations on a Binary Tree, Binary Tree Traversals (recursive), Binary search tree, B-tree , B+ tree, AVL tree, Threaded binary tree.	7
5	Graphs: Basic concepts, Different representations of Graphs, Graph Traversals (BFS & DFS), Minimum Spanning Tree(Prims &Kruskal) Dijkstra's shortest path algorithms. Hashing: Hash function, Address calculation techniques, Common hashing functions, Collision resolution: Linear and Quadratic probing, Double hashing.	8
TOTAL		40

3IT4-06 : Object Oriented Programming

Credit- 3
3L+0T+0P

Max. Marks : 150 (IA:30, ETE:120)
End Term Exam: 3 Hours

COURSE ASSESMENT METHODS: Two mid-term and one end-term examination	
PRE-REQUISITES: Basic knowledge of structured programming	
COURSE OBJECTIVE:	
1.	To understand the different programming paradigm
2.	To know the principles of OOPs
3.	Understand and apply the principles of inheritance
4.	Understand the principles of polymorphism
5.	Finally the objective of this course is to develop programming skills of undergraduate students to solve basic real world problems using objective oriented programming techniques.

SN	CONTENTS	Hours
1	Introduction to different programming paradigm, characteristics of OOP, Class, Object, data member, member function, structures in C++, different access specifiers, defining member function inside and outside class, array of objects.	8
2	Concept of reference, dynamic memory allocation using new and delete operators, inline functions, function overloading, function with default arguments, constructors and destructors, friend function and classes using this pointer.	8
3	Inheritance, types of inheritance, multiple inheritance, virtual base class, function overriding, abstract class and pure virtual function	9
4	Constant data member and member function, static data member and member function, polymorphism, operator overloading, dynamic binding and virtual function	9

5	Exception handling, Template, Stream class, File handling.	6
TOTAL		40

3IT4-07 : Software Engineering

Credit- 3
3L+0T+0P

Max. Marks : 150 (IA:30, ETE:120)
End Term Exam: 3 Hours

COURSE ASSESSMENT METHOD: Two midterm and one end term examination	
PRE-REQUISITES: Basic understanding of programming theory and applications.	
COURSE OBJECTIVES:	
1.	To study fundamental concepts in software engineering, SDLC, software requirements specification, formal requirements specification and verification.
2.	To study basic concepts of object-oriented design using UML, function-oriented software design. Design pattern and user interface design.
3.	To study the basic techniques for improving quality of software.
4.	Understand the fundamental principles of Software Project management & will also have a good knowledge of responsibilities of project manager and how to handle these.
5.	To understand the basic knowledge of Estimation model.

SN	CONTENTS	Hours
1	Introduction, software life-cycle models, software requirements specification, formal requirements specification, verification and validation.	8
2	Software Project Management: Objectives, Resources and their estimation, LOC and FP estimation, effort estimation, COCOMO estimation model, risk analysis, software project scheduling.	8
3	Requirement Analysis: Requirement analysis tasks, Analysis principles Software prototyping and specification data dictionary, Finite State Machine (FSM) models. Structured Analysis: Data and control flow diagrams, control and process specification behavioral modeling	8
4	Software Design: Design fundamentals, Effective modular design: Da	8

	architectural and procedural design, design documentation.	
5	Object Oriented Analysis: Object oriented Analysis Modeling, Data modeling. Object Oriented Design: OOD concepts, Class and object relationships, object modularization, Introduction to Unified Modeling Language .	8
	TOTAL	40

3IT4-21 : Data Structures and Algorithms Lab

Credit- 1.5
OL+OT+3P

Max. Marks : 75 (IA:45, ETE:30)
End Term Exam: 3 Hours

COURSE ASSESMENT METHODS: Internal Viva, Quiz and end term Practical examination	
PRE-REQUISITES: Basic knowledge of computer hardware	
COURSE OBJECTIVE:	
1.	To implement concepts of linear data structures like Stack and Queue.
2.	To implement various Sorting and Searching Techniques.
3.	To learn programming solve problems using both arrays and pointers.
4.	To implement concepts of non-linear data structures
5.	To implement hierarchical concept using tree.

SN	CONTENTS
1	Write a simple C program on a 32 bit compiler to understand the concept of array storage, size of a word. The program shall be written illustrating the concept of row major and column major storage. Find the address of element and verify it with the theoretical value. Program may be written for arrays up to 4-dimensions.
2	Simulate a stack, queue, circular queue and dequeue using a one dimensional array as storage element. The program should implement the basic addition deletion and traversal operations.
3	Represent a 2-variable polynomial using array. Use this representation to implement addition of polynomials
4	Represent a sparse matrix using array. Implement addition and transposition operations using the representation.
5	Implement singly, doubly and circularly connected linked lists illustrating operations like addition at different locations, deletion from specified locations and traversal.

6	Repeat exercises 2, 3 & 4 with linked structure.
7	Implementation of binary tree with operations like addition, deletion, traversal.
8	Depth first and breadth first traversal of graphs represented using adjacency matrix and list.
9	Implementation of binary search in arrays and on linked Binary Search Tree.
10	Implementation of different sorting algorithm like insertion, quick, heap, bubble and many more sorting algorithms.

3IT4-22 : Object Oriented Programming Lab

Credit-1.5
OL+OT+3P

Max. Marks : 75 (IA:45,ETE:30)
End Term Exam: 3 Hours

COURSE ASSESMENT METHODS: Internal Viva, Quiz and end term Practical examination	
PRE-REQUISITES: Basic knowledge of OOPS concepts and C	
COURSE OBJECTIVE:	
1.	To implement the concept of classes and objects
2.	To implement inheritance, polymorphism, encapsulation etc.
3.	To learn C++ programming by implementing various OOPS concepts
4.	To implement the file handling programs

SN	CONTENTS
1	Understand the basics of C++ library, variables, data input-output.
2	C++ program using with the concept of structures.
3	Implement class and object concepts and function overloading.
4	Write programs to understand dynamic memory allocation and array of objects.
5	Program to understand different types of constructors and destructor.
6	Implement friend function to access private data of a class and usage of this pointer.
7	Write programs to understand the usage of constant data member and member function, static data member and member function in a class.
8	Implement different types of inheritance, function overriding and virtual function
9	Implement Operator overloading concepts.
10	Write programs to understand function template and class template.
11	Write programs to understand exception handling techniques.
12	Write programs to understand file handling techniques.

3IT4-23 : Software Engineering Lab

Credit- 1.5
OL+OT+3P

Max. Marks : 75 (IA:45, ETE:30)
End Term Exam: 3 Hours

COURSE ASSESSMENT METHOD: Internal Viva, Quiz and end term Practical examination	
PRE-REQUISITES: Basic understanding of Software Engineering Concepts	
COURSE OBJECTIVES:	
1.	To impart the knowledge of Software Requirement Specification (SRS).
2.	To understand the destine structure of DFD model with documents.
3.	To develop the knowledge of structured design for DFD.
4.	To understand the behavior of structure UML diagrams.
5.	To learn about ProjectLibre project management software tool.

SN	CONTENTS
1	Development of requirements specification, function oriented design using SA/SD, object-oriented design using UML, test case design, implementation using Java and testing. Use of appropriate CASE tools and other tools such as configuration management tools, program analysis tools in the software life cycle.
2	Develop Software Requirements Specification (SRS) for a given problem in IEEE template.
3	Develop DFD model (level-0, level-1 DFD and Data dictionary) of the project.
4	Develop structured design for the DFD model developed.
5	Developed all Structure UML diagram of the given project.
6	Develop Behavior UML diagram of the given project.
7	Manage file, using ProjectLibre project management software tool.

3IT4-24 : Digital Electronics Lab

Credit- 1.5
OL+OT+3P

Max. Marks : 75 (IA:45, ETE:30)
End Term Exam: 3 Hours

COURSE ASSESSMENT METHOD: Internal Viva, Quiz and end term Practical examination	
PRE-REQUISITES: Basic understanding of Electrics and Electronics	
COURSE OBJECTIVES:	
1.	To acquire the basic knowledge of digital logic.
2.	Application of knowledge to understand digital electronic circuit.
3.	To prepare students perform the analysis of various digital electronic circuits.
4.	To develop skills to build and troubleshoot digital circuits.

SN	CONTENTS
1	To verify the truth tables of basic logic gates: AND, OR, NOR, NAND, NOR. Also to verify truth table of Ex-OR, Ex-NOR (For 2, 3, & 4 inputs using gates with 2, 3 & 4 inputs).
2	To verify the truth table of OR, AND, NOR, Ex-OR, Ex-NOR realized using NAND & NOR gates.
3	To realize an SOP and POS expression.
4	To realize Half adder/ Subtractor & Full Adder/ Subtractor using NAND & NOR gates and to verify their truth tables.
5	To realize a 4-bit ripple adder/ Subtractor using basic Half adder/ Subtractor & basic Full Adder/ Subtractor.
6	To verify the truth table of 4-to-1 multiplexer and 1-to-4 demultiplexer. Realize the multiplexer using basic gates only. Also to construct and 8-to-1 multiplexer and 1-to-8 demultiplexer using blocks of 4-to-1 multiplexer and 1-to-4 demultiplexer.
7	Design & Realize a combinational circuit that will accept a 2421 BCD code and

	drive a TIL -312 seven-segment display.
8	Using basic logic gates, realize the R-S, J-K and D-flip flops with and without clock signal and verify their truth table.
9	Construct a divide by 2,4& 8 asynchronous counter. Construct a 4-bit binary counter and ring counter for a particular output pattern using D flip flop.
10	Perform input/output operations on parallel in/Parallel out and Serial in/Serial out registers using clock. Also exercise loading only one of multiple values into the register using multiplexer. Note: As far as possible, the experiments shall be performed on bread board. However, experiment Nos. 1-4 are to be performed on bread board only.