SECOND YEAR: INFORMATION TECHNOLOGY

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER –III

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
IT 3.1	Applied Mathematics-III	3	1	--	3	100	25	--	--	--	**125**
IT 3.2	Numerical Methods	3	1	2	3	100	25	25	--	--	**150**
IT 3.3	Signals and Systems	3	1	--	3	100	25	--	--	--	**125**
IT 3.4	Analog and Digital Circuits	3	1	2	3	100	25	--	--	25	**150**
IT 3.5	Data Structures	3	1	2	3	100	25	--	25	--	**150**
IT 3.6	Object-Oriented Programming System	3	1	2	3	100	25	--	25	--	**150**
TOTAL		**18**	**06**	**08**	**--**	**600**	**150**	**25**	**50**	**25**	**850**

A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

SECOND YEAR: INFORMATION TECHNOLOGY

SCHEME OF INSTRUCTION AND EXAMINATION

(RC 2016-17)

SEMESTER –IV

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P#	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
IT 4.1	Discrete Mathematical Structures	3	1	--	3	100	25	--	--	--	**125**
IT 4.2	Entrepreneurship Development	3	--	--	3	100	25	--	--	--	**125**
IT 4.3	Computer Organization and Architecture	3	1	2	3	100	25	25	--	--	**150**
IT 4.4	Software Engineering	3	1	2	3	100	25	--	25	--	**150**
IT 4.5	Design and Analysis of Algorithms	3	1	2	3	100	25	--	--	25	**150**
IT 4.6	Microprocessors and Interfacing	3	1	2	3	100	25	--	25	--	**150**
TOTAL		**18**	**05**	**08**	**--**	**600**	**150**	**25**	**50**	**25**	**850**

A candidate is considered to have successfully fulfilled the requirement of a semester, provided he/ she submits to the department a certified journal reporting the experiments conducted during the semester.

IT 3.1 Applied Mathematics-III

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
IT 3.1	Applied Mathematics-III	3	1	--	3	100	25	--	-	--	**125**

Course Objectives: The aim of learning this course is to provide students with the mathematical knowledge and skills necessary to support their concurrent and subsequent engineering studies.

Course Outcomes:

After successful completion of this course the student would be able to

1. Compute the rank and inverse of a matrix and solve system of linear equations.
2. Compute Eigen values and Eigen vectors of a given matrix, apply Cayley Hamilton theorem.
3. Understand the basic concepts of probability, random variables, mean, variance, standard deviation and probability distributions, correlation and regression.
4. Use tools like Laplace transforms and Fourier transforms in formulating and solving Engineering problems.
5. Understand Z- transforms and its properties and apply it in solving difference equations.

UNIT - 1 (12 Hours)

Linear Algebra: Types of matrices, adjoint, inverse. Elementary transformations.Rank of a matrix, normal form, echelon form. Linear system of equations AX = B and AX = 0. Linearly independent and dependent vectors, Eigen values and Eigen vectors, Cayley Hamilton Theorem, minimal equation, Diagonalization.

UNIT - 2 (14 Hours)

Probability and Probability distributions: Definition, properties, Axioms of probability, Conditional probability, Baye's theorem, Random Variables. Discrete probability distribution, Continuous probability distribution, Distribution function. Expectation and Variance, Moment generating function. Special distributions: Binomial, Poisson, Geometric, Normal, Uniform and exponential. Correlation and regression.

UNIT - 3 (10 Hours)

Laplace Transforms: Definition, Existence conditions, properties, inverse Laplace

transforms. Laplace transform of periodic functions, Convolution theorem, Laplace
transform of Dirac-Delta function, Application of Laplace transforms in solving linear
differential equations with initial conditions and system of linear simultaneous differential equations.

UNIT - 4 **(12 Hours)**

Fourier and Z-transforms: Definition, properties, inverse. Convolution theorem. Applications of Fourier and Z-transforms.

Recommended Readings:

1. Grewal B. S.; Higher Engineering Mathematics; Khanna Publications, New Delhi.
2. H. K. Dass; Advanced Engineering Mathematics; S. Chand &Co.
3. Erwin Kreyzing; Advanced Engineering Mathematic; Wiley.
4. Kandasamy, P.; Engineering Mathematics; Chand & Co., New Delhi.
5. Srimanta Pal, Subodh C. Bhunia; Engineering Mathematics; Oxford University Press.
6. Dr. D. S. C ; Engineering Mathematics- Part III ; Prism Books Pvt. Ltd.
7. Montgomery, D. C., Probability and Statistics for Engineers; Prentice Hall of India.

IT 3.2 NUMERICAL METHODS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
IT 3.2	Numerical Methods	3	1	2	3	100	25	25	--	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An ability to understand the use of numerical methods in modern scientific computing,
2. An understanding of finite precision computation
3. An understanding to calculation and interpretation of errors in numerical method.

Course Outcomes:

The student after undergoing this course will be able to:

1. Apply numerical algorithms to solve Engineering problems.
2. Explain the use of numerical methods in modern scientific computing.
3. Design algorithms with finite precision.
4. Explain the calculation and interpretation of errors in numerical methods.
5. Illustrate the working of numerical algorithms using examples.

UNIT -1 (12 Hours)

Errors and Approximations: Introduction, sources of errors, problems in computations, safeguards against errors, floating point arithmetic, absolute error, relative error, percentage error – calculations. Solutions of Non-linear equations :Bisection Method, False Position Method, Newton Raphson, Secant methodDirect solution of Linear Equations: Solution by Elimination, Basic Gauss Elimination method, Gauss Elimination with pivoting, Gauss – Jordan method

UNIT -2 (12 Hours)

Iterative Solutions of Linear Equations: Jacobi iteration method, Gauss Seidel method, Method of relaxation, convergence of iteration methods. Interpolation: Linear Interpolation, Lagranges Interpolation Polynomial, Newton's Interpolation Polynomial, Divided difference table, Interpolation with Equidistant points. Extrapolation, Inverse interpolation. Regression: Fitting Linear Equations, Fitting transcendental equations, Fitting polynomial function

UNIT -3 (12 Hours)

Numerical Differentiation: Differentiating Continuous Functions, Differentiating Tabulated functions, difference tables, Richardson Extrapolation.
Numerical Integration: Trapezoidal Rule, Simpson's 1/3 rule, Simpson's 3/8 rule, Romberg Integration

UNIT -4 (12 Hours)

Numerical Solution of Ordinary Differential equations: Taylor Series Method, Euler's methods, Heun's Method, Polygon Method, Runge-Kutta methods. Numerical Solution of Partial Differential Equations: Deriving differential Equations, Elliptic Equations, Parabolic Equations, Hyperbolic Equations .

Recommended Readings:

1. E. Balaguruswamy; Numerical Methods;Tata Mc Graw Hill.
2. S. S. Shastry ; Introductory Methods of Numerical Analysis; PHI.
3. E.V. Krishnamurthy and Sen; Numerical Algorithms; PHI.
4. Rajaraman; Computer Oriented Numerical Techniques; PHI.
5. B.S. Grewal ; Numerical Methods in Engineering and Science; Khanna Publications.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. The Term Work Marks to be awarded based on the assessment of experiments conducted.)

1. To calculate absolute, relative and percentage relative error.
2. To implement Bisection method.
3. To implement Regula Falsi method.
4. To implement Newton Raphson method.
5. To implement Secant method.
6. To implement Gauss Seidal method.
7. To implement Lagranges interpolation.
8. To implement Trapezoidal rule.
9. To implement Simpson's rule.
10. To implement Euler's method.

IT 3.3 SIGNALS AND SYSTEMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
IT 3.3	Signals and Systems	3	1	--	3	100	25	--	--	--	**125**

Course Objectives:

The subject aims to provide the student with:

1. Understanding of time-domain representation and analysis of signals and systems.
2. An ability to perform frequency-domain representation and analysis using Fourier tools.
3. An ability to perform frequency-domain representation and analysis using Laplace transform and Z transforms.
4. An understanding of sampling, aliasing and Signal reconstruction.

Course Outcomes:

The student after undergoing this course will be able to:

1. Classify different types of signals and systems.
2. Illustrate the properties of continuous-time and discrete-time systems.
3. Analyze Continuous-time (CT) and discrete-time (DT) systems in time-domain using convolution.
4. Analyze CT and DT systems in Frequency domain using tools like CTFS, CTFT, DTFS and DTFT.
5. Explain the concepts of Sampling, aliasing and Signal reconstruction.
6. Analyze CT and DT systems using Laplace transforms and Z Transforms.

UNIT -1 **(12 Hours)**

Introduction to signals and systems. Overview of Specific Systems, Classification of Signals, Basic operations on Signals, Elementary Signals, Systems Viewed as Interconnection of operations, Properties of System.

Time-Domain Representation of Linear Time-Invariant Systems: Introduction, The Convolution Sum, Convolution Sum Evaluation Procedure, The Convolution Integral, Convolution Integral Evaluation Procedure, Interconnections of LTI systems, Relations between LTI system Properties and Impulse Response, Step Response.

UNIT -2 **(12 Hours)**

Fourier Representation of Signals and Linear Time-Invariant Systems – Complex Sinusoids and Frequency Response of LTI Systems, Fourier Representations for Four Classes of Signals. The Fourier Transform, Properties of Fourier Representations, Linearity, Symmetry, Convolution, Differentiation and Integration, Time and Frequency –Shift , Finding Inverse Fourier Transforms by Using Partial-Fraction Expansions, Multiplication Property, Scaling Properties, Parsevals Relationships, Time-Bandwidth Product, Duality.
Applications of Fourier Representations: a) Mixed Signal Classes - Fourier Transform Representations of Periodic Signals, Convolution and Multiplication with Mixtures of Periodic and Non periodic Signals, Fourier Transform Representation of Discrete-Time Signals, Sampling, Reconstruction of Continuous-Time Signals from b) Communication Systems –Type of Modulation, Benefits of Modulation, Full Amplitude Modulation.

UNIT -3 **(12 Hours)**

Representing Signals by Using Continuous-Time Complex Exponentials: the Laplace Transform – Introduction, The Laplace Transform, The Unilateral Laplace Transform, Properties of the Unilateral Laplace Transform, Inversion of the Unilateral Laplace Transform, Properties of the Bilateral Laplace Transform, Properties of the Region of Convergence, Inversion of the Bilateral Laplace Transform, The Transfer Function, Causality and Stability, Determining the Frequency Response from Poles and Zeros.

UNIT -4 **(12 Hours)**

Representing Signals by Using Discrete-Time Complex Exponentials: the z-Transform – Introduction, The z-Transform, Properties of the Region of Convergence, Properties of the z-Transform, Inversion of the z-Transform, The Transfer function, Causality and Stability, Determining the Frequency Response from Poles and Zeros, Computational Structures for Implementing Discrete-Time LTI Systems, The Unilateral z-Transform.

Recommended Readings:

1. Simon Haykin and Barry Van Veen; Signals and Systems; John Wiley &Sons (Asia)Pvt. Ltd; 2/e.
2. Oppenheim and Willskay with Hamid Nawab; Signals and Systems; Prentice Hall of India.
3. Linder; Introduction to Signals and Systems; McGraw Hill.
4. Nagrath, Sharan, Rajan and Kumar; Signals and Systems; McGraw Hill.
5. Zeimer, Tranter, Fannin, IE; Signals and Systems; Prentice Hall of India.

IT 3.4 ANALOG AND DIGITAL CIRCUITS

<table>
<tr><th rowspan="3">Subject Code</th><th rowspan="3">Name of the Subject</th><th colspan="3">Scheme of Instruction Hrs/Week</th><th colspan="7">Scheme of Examination</th></tr>
<tr><th rowspan="2">L</th><th rowspan="2">T</th><th rowspan="2">P</th><th rowspan="2">Th Duration (Hrs)</th><th colspan="6">Marks</th></tr>
<tr><th>Th</th><th>S</th><th>TW</th><th>P</th><th>O</th><th>Total</th></tr>
<tr><td>IT 3.4</td><td>Analog and Digital Circuits</td><td>3</td><td>1</td><td>2</td><td>3</td><td>100</td><td>25</td><td>--</td><td>--</td><td>25</td><td>150</td></tr>
</table>

Course Objectives:

The subject aims to provide the student with:

1. An understanding of various Number Systems & Codes along with Boolean algebra.
2. An ability to solve Boolean algebra problems.
3. An ability to design combinational and sequential circuits.
4. An understanding of integrated analog and digital circuits.
5. An understanding of amplifiers, voltage regulators and oscillators.

Course Outcomes:

The student after undergoing this course will be able to:

1. Convert the numbers from one radix to another and perform arithmetic operations using the 1's and 2's compliments.
2. Solve Boolean Expressions using Boolean algebra, K-maps and VEM and implement them using logic gates.
3. Design any given combinational circuits.
4. Explain different flip flops, registers and their applications.
5. Design sequential circuits and state machines.
6. Design synchronous and asynchronous counter circuits.
7. Explain arithmetic circuits like adders and multipliers and their applications.
8. Design timer and analog and digital circuits using op amps.

<u>UNIT -1</u> **(12 Hours)**

Digital Logic: Binary Numbers, basic gates, Boolean algebra, Nor and Nand Gates, And or Invert Gates, De Morgan's theorem, Positive and Negative Logic.

Arithmetic Circuits Binary Addition & Subtraction, Unsigned binary numbers, 2's Complement Representation & Arithmetic, Adder-Subtracter.

Combination Logic Circuits Boolean laws/theorems, Sum of Products, Truth table, Pairs, Quads, and Octets, Karnagh mapping, Product of Sums Method and Simplification.

UNIT -2 (12 Hours)

Data Processing Circuits: Multiplexers, Demultiplexers, decoder, BCD to decimal decoder, 7-segmant decoder, encoders. Flip-Flops: RS Flip-Flops , D and JK Flip-Flops, Flip-Flop timing, JK Master-Slave Flip-Flops. Timing Circuits: Schmitt Trigger, 555 Timers Astable, Monostables with input logic.

UNIT -3 (12 Hours)

Registers: Types of Registers, Serial in-serial out, Serial in-parallel out, Parallel in-serial out, Parallel in-parallel out, Ring counters.

Counters: Asynchronous counters, Synchronous counters, changing the counter modulus, decade, and shift counters, A MOD-10 shift counter with decoding. D/A and A/D conversion and its Specifications.

UNIT -4 (12 Hours)

Op-amp: ideal characteristics ,op-amp-as inverting amplifier – op-amp as non-inverting amplifier.
Application: Adder, Subtractor, Integrator, Differentiator, Comparator
Oscillators: Barkhansen criterion for oscillation – Hartley oscillator – Colpits' oscillator phase shift oscillator Piezoelectric crystals – crystal oscillator.
Voltage Regulators: Definition, design and letter using IC 723.

Recommended Readings:

1. R.P. Jain; Modern Digital Electronics; TMH; 2/e.
2. Ramakant A. Gayakwad ; OpAmps& Linear Integrated Circuits; PHI; 2/e.
3. N NBhargava, D C Kulshreshtha, S C Gupta ; Basic Electronics and Linear Circuits; McGraw Hill Education (India) Private Limited;2/e.
4. A.P. Malvino, Donald P. Leach ; Digital Principles and Applications; TMH; 4/e.
5. Malvino ; Digital Computer Electronics; TMH; 2/e.
6. MIllman and Halkias ; Integrated Electronics: Analog and Digital Electronic Circuits and Systems; TMH.
7. Vishwanathan, Mehta and Rajaraman ; Electronics for Scientist & Engineer; PHI.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Verification of Logic Gates.
2. Boolean Algebra.
3. BCD to Excess 3 & Excess 3 to BCS Conversion.
4. Binary to Gray code and Gray code to Binary Conversion.

5. Encoder and Decoder.
6. Mux and De-mux using only NAND gates.
7. Flip Flops.
8. Johnson Counter / Ring Counter.
9. Sequence Generator.
10. Multivibrator.

IT 3.5 DATA STRUCTURES

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
IT 3.5	Data Structures	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

1. Demonstrate understanding of the abstract properties of various data structures such as stacks, queues, lists, trees and graphs
2. Use various data structures effectively in application programs.
3. Demonstrate understanding of various sorting algorithms
4. Understand and apply fundamental algorithmic problems including Tree traversals, Graph traversals, and shortest paths.
5. Demonstrate understanding of various searching algorithms.

Course Outcomes:

The student after undergoing this course will be able to:

1. Use different data structures.
2. Describe common applications for arrays, linked lists, stacks, queues, trees, and graphs.

UNIT -1 (12 Hours)

Introduction to data representation and data structures. Representation of arrays and their applications. Stacks: representation of stacks and its applications, Recursion, Tower of Hanoi, Implementation of recursive procedures by stacks

Queues: representation of queues and its applications, circular queues, priority queues, dequeue.

UNIT -2 (12 Hours)

Lists: Singly linked list, doubly linked list, circular linked list, linked stacks and queues and its applications.

Trees: Basic terminology, binary trees and their representations, traversals of trees, applications of trees – infix/postfix representation if expressions and inter-conversion, etc. B-tree, AVL.

UNIT -3 (12 Hours)

Sorting: Basic concept, Exchange sort, Selection sort, Insertion sort, Quick sort, Tree sort, Merge sort, Radix sort, Heaps and Heap sort.

Searching: Basic searching techniques, sequential and binary search, tree searching

Hashing: Hash function, collision handling mechanisms.

UNIT -4 (12 Hours)

Graphs: Basic terminology, representation of graphs, directed and undirected graphs and their traversals, depth first and breadth first search, spanning trees Applications of graphs: shortest path problem, topological sorting, matching.

Recommended Readings:

1. Alfred V. Aho, John E. Hopcroft & J. D. Ullman; Data Structures and Algorithms; Addison Wesley.
2. Yedidyah Langson, Moshej Augenstein, Aaron M. Tenenbaum; Data Structures using C & C++; Prentice Hall of India.
3. Robert L. Kruse; Data Structures and Program Design in ; PHI.
4. Sahni; Data Structures, Algorithms and Applications in C++; MGH.
5. Ellis Horowitz and Sartaj Sahni ; Fundamentals of Data Structures; Galgotia Publications.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Implement stack and use it to convert infix to postfix expression.
2. Linked implementation of stack.
3. Array implementation of queue.
4. Linked implementation of queue.
5. Array based circular queue.
6. Implement dequeue using array.
7. Implement singly linked lists.
8. Implement doubly linked lists.
9. Implement an binary tree. Produce its pre-order, in-order, and post-order traversals.
10. Implement binary search tree.
11. Implement hashing techniques.
12. Implement the following sorting algorithms:
    Bubble Sort, insertion sort, selection sort, heap sort, quick sort, merge sort.
13. Implement linear search and binary search.
14. Breadth first search and Depth first search.

IT 3.6 OBJECT ORIENTED PROGRAMMING SYSTEM

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
IT 3.6	Object Oriented Programming Systems	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of the concept of object oriented programming.
2. An understanding of the concepts of data hiding, data abstraction, polymorphism inheritance and exception handling.
3. Ability to understand the generic principles of object oriented programming using "C++".
4. An understanding the use of templates in "C++".
5. An ability to plan, design, execute and document sophisticated object oriented programs to handle different computing problems.

Course Outcomes:

The student after undergoing this course will be able to:

1. Differentiate between structure oriented programming and object oriented programming.
2. Design algorithms using principles of object oriented programming.
3. Apply concepts of operator overloading, constructors and destructors.
4. Explain the applications of polymorphism and inheritance in object oriented programming.
5. Apply the knowledge of standard template library achieve reusability.
6. Illustrate stream I/O and exception handling.

<u>UNIT - 1</u> **(12 Hours)**

Introduction: Principles of object oriented programming, object-oriented paradigm. Overview and Benefits of Object-Oriented Programming. Basic concept of oops-Objects, Classes, Encapsulation, Data Abstraction, Inheritance, Polymorphism, Dynamic Binding, Message Passing. Structure of a C++ program, Data types. Constants , tokens, expressions, control structures, functions ,arrays, Strings.

<u>UNIT - 2</u> **(12 Hours)**

Classes and Objects, Constructors and destructors. Concepts of polymorphism, Function overloading, operator overloading, Overloading types, & rules, explicit & implicit type conversion operators, Pointers and Pointers arithmetic.

UNIT - 3 (12 Hours)

Inheritance, extending classes, multiple inheritance, hybrid inheritance, pointers, virtual functions, and classes, and polymorphism. I/O streams and classes, Manipulators, Classes for file streams, file I/O operations and functions.

UNIT - 4 (12 Hours)

Template functions and classes, implementation, Exception handling: Need, Throwing mechanism, try, catch block, Introduction to the Standard Template Library: Components of STL, Containers, Algorithms, Iterators , Applications.

Recommended Readings:

1. E Balaguruswamy; Object oriented programming with C++; Tata McGraw Hill.
2. K R Venugopal, Rajkumar, T. Ravishankar; Mastering C++; Tata McGraw Hill.
3. Paul Deitel, Harvey M. Deitel; C++ for Programmers; Pearson Education.
4. Herbert Schildt ; Teach yourself C++; TMH.
5. J. R. Hubbar; Programming with C++; Schaum's Outlines;McGraw Hill.
6. D. Ravichandran ; Programming with C++; McGraw Hill.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Basics of C++ (input /output / control statements / functions/ array/ string)
2. Classes and Objects
3. Constructors and Destructors with Pointers.
4. Operator Overloading.
5. Inheritance and Polymorphism.
6. Console I/O and files.
7. Template.
8. Exception Handling.
9. STL and String manipulation.
10. Mini Project (Individual project).

IT 4.1 Discrete Mathematical Structures

<table>
<tr><th rowspan="3">Subject Code</th><th rowspan="3">Name of the Subject</th><th colspan="3">Scheme of Instruction Hrs/Week</th><th colspan="7">Scheme of Examination</th></tr>
<tr><th rowspan="2">L</th><th rowspan="2">T</th><th rowspan="2">P</th><th rowspan="2">Th Duration (Hrs)</th><th colspan="6">Marks</th></tr>
<tr><th>Th</th><th>S</th><th>TW</th><th>P</th><th>O</th><th>Total</th></tr>
<tr><td>IT 4.1</td><td>Discrete mathematical Structures</td><td>3</td><td>1</td><td>--</td><td>3</td><td>100</td><td>25</td><td>--</td><td>-</td><td>--</td><td>125</td></tr>
</table>

Course Objective: This course is designed to introduce students to the techniques, algorithms, and reasoning processes involved in the study of discrete mathematical structures that are essential to the field of Computer Science.

Course Outcomes: On completing this course students will be able to

1. Perform operations on discrete structures such as sets, functions, relations, and sequences.
2. Know the properties of equivalence relations and partial orderings.
3. Apply algorithms and use definitions to solve problems to prove statements in elementary number theory.
4. Construct mathematical arguments using logical connectives and quantifiers and verify the correctness of an argument using propositional and predicate logic and truth tables.
5. solve problems using the basic principles of counting theory, including permutation, combinations, and the pigeonhole principle
6. Solve problems involving recurrence relations and generating functions.
7. Understand lattices and Boolean algebras.
8. Explain basic definitions and properties associated with simple planar graphs, including isomorphism, connectivity, and Euler's formula, and describe the difference between Eulerian and Hamiltonian graphs.
9. Use graphs and trees as tools to solve combinatorial optimization problems

UNIT - 1 **(12 Hours)**

Set Theory : Sets, Set Operations, Relations and their properties, Equivalence Relations, partial orderings.
Functions: One-to-One and Onto Functions, Inverse Function, Composition of functions, Graphs of functions and some important functions.
Integers: Integers and division (excluding applications of congruences and cryptology), primes and greatest common divisors, Integers and algorithms.

UNIT - 2 (12 Hours)

Propositional Calculus: Propositional logic, propositional equivalences, predicates and quantifiers, rules of inference.
Boolean Algebra: Boolean functions, representing Boolean functions.
Mathematical Induction: Principle of Mathematical Induction and applications.

UNIT - 3 (12 Hours)

Counting: The basics of counting, pigeonhole principle, permutations and combinations, binomial coefficients.
Advanced Counting Techniques: Recurrence relations, solving linear recurrence relations, inclusion –exclusion principle, applications of inclusion – exclusion principle.

UNIT - 4 (12 Hours)

Graph theory: Graphs and graph models, graph terminology and special types of graphs, representing graphs and graph isomorphism, connectivity, Euler and Hamilton paths, shortest path problems, planar graphs, graph coloring.
Trees: Introduction to Trees, applications of trees, tree traversal, Spanning Trees, Minimal Spanning Trees.

Recommended Readings:

1. Kenneth H. Rosen; Discrete Mathematics and Its Applications; Tata McGraw Hill (6th edition).
2. B Kolman, R.C. Busby and Sharon C. Ross; Discrete Mathematical Structures; Prentice Hall.
3. J. P. Tremblay and R. Manohar, McGraw Hill; Discrete Mathematical Structures with Applications to Computer Science; New York McGraw Hill.
4. Swapan Kumar Sarkar; Discrete Mathematics; S.Chand Publication.
5. Dr. D. S. C ;Discrete Mathematical Structures; Prism Books Pvt. Ltd.
6. G.V.Kumbhojkar; Discrete Structures And Graph Theory; Pradeep Prakashan.

IT 4.2 ENTREPRENEURSHIP DEVELOPMENT

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
IT 4.2	Entrepreneurship Development	3	1	--	3	100	25	--	--	--	**125**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of qualities and requirements of an entrepreneur.
2. An ability to understand the requirements of Project identification, development and implementation.
3. An understanding of Break even analysis.
4. An understanding of the role of Communication in organizations.
5. An understanding of the complexity of managing in a global world.

Course Outcomes:

The student after undergoing this course will be able to:

1. Demonstrate the skills for project identification, development and implementation.
2. Explain the essential qualities and requirements of an entrepreneur.
3. Apply the concepts of Break even analysis.
4. Explain the role of effective Communication in organizations.
5. Apply managerial concepts to solve complex problems related to global issues.

<u>UNIT - 1</u> **(12 Hours)**

Definition and clarification of concept of entrepreneurship: Qualities of an entrepreneur
Skills required for entrepreneurship, Functions of an entrepreneur, Importance of entrepreneur in economic development. Theories of Entrepreneurship: Economic theory, Sociological theory, Psychological theory. Types of entrepreneurs: Based on type of business, Based on use of technology, Based on motivation, Based on stages of development, Based on motive, Based on capital ownership, Danhof's classification. Project identification: External environment analysis, Meaning and characteristics of a project, Classification of projects, Project life-cycle, Project identification, Sources and screening of project ideas.

Project formulation: Meaning and significance, Feasibility analysis, Techno-economic analysis, Input analysis, Financial analysis, Social cost benefit analysis. Project feasibility. Pre-feasibility study: Project feasibility report - Meaning, Importance and Contents. Importance of location of a project.

UNIT - 2 (12 Hours)

Project financing and institutional finance: Classification of capital – Fixed capital -Meaning, Factors governing fixed capital requirements, Working capital – Meaning and concepts, Types, Factors determining working capital requirements. Sources of finance – Share capital, Debenture capital, Lease finance and term loans from commercial banks. Institutional finance. IFCI, ICICI, IDBI, SIDBI, EXIM Bank, Commercial banks – Functions and schemes. Small scale industries: Definition and characteristics, Role in Indian economy, Steps for starting a SSI unit, Problems faced by SSIs. Incentives and subsidies – Need and Types.

UNIT - 3 (12 Hours)

Financial aspects: Break even analysis, Income statement, Balance sheet. Profit and loss account, Fund flow statement, Ratio analysis – Liquidity, leverage and profitability ratios. Capital budgeting – Need, Importance, Process, Nature of capital budgeting problem, Weighted average cost of capital, approaches to fixing a capital budget, methods of project evaluation: Payback period, Accounting rate of return, discounted cash flow, Net Present Value Index.

UNIT - 4 (12 Hours)

Managerial aspects: Introduction to management, Functions of a manager, Different schools of management. Types of organisation structures, Leadership-Trait theory, Behavioural theory, Contingency theory, Motivation -Carrot and stick theory, Maslow's theory, Herzberg's theory, Vroom's theory, McClelland's theory. Communication – Importance, Process, types and forms, Barriers to communication, Principles of effective communication. Marketing management, Meaning and importance, Marketing mix, Types of marketing tasks, Market segmentation – process and criteria, Marketing implementation and control.

Recommended Readings:

1. A. Vinod ; Entrepreneurial Development and Project Management ; Calicut University Publication; 2002, 4/e.
2. C.B. Gupta and S.S. Khanka; Entrepreneurship and Small Business Management; Sultan Chand and Sons; 1997,2/e.
3. C.B.Gupta and N.P.Srinivasan ; Entrepreneurship; Sultan Chand and Sons; 1997,4/e.
4. Philip Kotler ; Marketing Management ; Pearson Education, 2003; 11/e.
5. P. C. Tripathi and P.N. Reddy ; Principles of Management ; Tata McGraw Hill; 1991, 2/e.
6. Prassanna Chandra; Fundamentals of Financial Management; Tata McGraw Hill; 2001, 3/e.
7. Harold Koontz and Heinz Weihrich ; Management; McGraw Hill; 1988, 9/e.

IT 4.3 COMPUTER ORGANIZATION AND ARCHITECTURE

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
IT 4.3	Computer Organization and Architecture	3	1	2	3	100	25	25	--	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of relationship between hardware and software.
2. An ability to recognize how machine organization impacts the efficiency of applications written in a high-level language.
3. An ability to understand the system performance and concepts of RISC architecture.
4. An understanding of different ways of communicating with I/O devices and standard I/O interfaces.
5. An understanding of memory hierarchy.
6. An ability to develop solutions for basic programs using assembly language.

Course Outcomes:

The student after undergoing this course will be able to:

1. Explain the organization of the Control unit, Arithmetic and Logical unit, Memory unit and the I/O unit.
2. Identify high performance architecture design.
3. Create an assembly language program to program a microprocessor system.
4. Explain the ways to take advantage of instruction level parallelism for high performance processor design.
5. Demonstrate memory hierarchy and its impact on computer cost/performance.

<u>UNIT - 1</u> (12 Hours)

Introduction: Organization and Architecture, Structure and function

A top level view of Computer Function and Interconnection: Computer Components, Computer Function, Interconnection structure, Bus interconnection.

Computer Arithmetic: Arithmetic and Logic Unit, Integer Representation, Integer Arithmetic, Floating Point Representation, Floating Point Arithmetic.

Instruction Sets: Machine Instruction Characteristics, Types of Operands, Addressing Modes and Formats- Addressing, Instruction Formats.

UNIT - 2 (12 Hours)

Cache Memory: Computer memory system Overview, Cache Memory Principles, Elements of Cache Design

Internal Memory Technology: Semiconductor Main Memory, Error correction, RAM and ROM Chips , Memory address map/Memory connection to CPU , Associative Memory -Hardware organization

External Memory: Magnetic Disk, RAID, Optical Memory

Virtual Memory : Address space and memory space, Address mapping using pages

Associative memory page table, Page replacement.

UNIT - 3 (12 Hours)

Input/output: External Devices, I/O Modules, Programmed I/O, Interrupt Driven I/O, Direct Memory Access (DMA Controller), I/O Channel and Processor.

Processor Structure and Functions: Processor Organization, Register Organization, The instruction cycle, Instruction Pipelining.

Quantitative principles of computer design : Amdahl's Law, CPU performance equation- MIPS as a metric.

UNIT - 4 (12 Hours)

Reduced Instruction Set Computers (RISCs) : Instruction execution characteristics, The use of Large Register file, Compiler based Register Organization, Reduced instruction set architecture, RISC versus CISC controversy, Introduction to VLIW.

Control unit operation: Micro operations, Control of the processor, Hardwired implementation.

Micro programmed control: Basic concepts, Microinstruction sequencing, Microinstruction execution.

Recommended Readings:

1. William Stalling; Computer Organization and Architecture: Designing for performance; Pearson Education; 2010; 8/e ; . ISBN 978-81-317-3245-8.
2. Morris Mano ; Computer system architecture; Pearson Education; 1993; 3/e; ISBN 81-7808-687-5.
3. Patterson and Hennessy; Computer Architecture A Quantitative Approach; Morgan Kaufmann Publishers; 1996; 2/e; ISBN 1-55860-329-8.
4. Kai Hwang; Advanced Computer Architecture - Parallelism, Scalability, Programmability; Tata McGraw Hill, 2010; 2/e.
5. Carl Hamacher, Zvonko Vranesic, Safal Zaky; Computer Organization; 5/e.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments. The Term Work Marks to be awarded based on the assessment of experiments conducted.)

1. Program to convert a decimal number into binary value.
2. Program to convert a decimal number into hexadecimal value.
3. Program to convert a decimal number into octal value.
4. Program to convert a binary number into decimal value.
5. Program to convert a hexadecimal number into decimal value.
6. Program to convert a octal number into decimal value.
7. Program to Implement Floating-Point Addition.
8. Program to Implement Floating-Point Subtraction.
9. Program to Implement Floating-Point Multiplication.
10. Program to Implement Floating-Point Division.
11. Program to Implement Multiplication of Unsigned Binary Integers.
12. Program to Implement Booth's Algorithm for Two's Compliment Multiplication.

IT 4.4 SOFTWARE ENGINEERING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
					Th Duration (Hrs)	Marks					
		L	T	P		Th	S	TW	P	O	Total
IT 4.4	Software Engineering	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An understanding of the current issues and practices in software engineering with an emphasis on the software development process.
2. An ability to understand the software planning and management.
3. Ability to plan software requirements specifications, system modeling, quality specifications, and program specifications.
4. An understanding of software design approaches.
5. An understanding of the requirements of software project management.
6. An ability to recognize social, ethical, cultural, and safety issues in software deployment.

Course Outcomes:

The student after undergoing this course will be able to:

1. Design a specification a software system for any existing system.
2. Plan a design of software system as per the specification.
3. Implement a software system it with readable, reusable, modular and object-oriented techniques.
4. Design a test procedure for validity, correctness and completeness.
5. Implement a software maintenance schedule.
6. Demonstrate the skills of a Software Designer, Software Architect or Project Manager for the development of software to solve business and technical problems.
7. Explain the methodologies, architectural approaches, project management techniques, and team dynamics.

<u>UNIT - 1</u> **(12 Hours)**

Introduction to Software Engineering, scope of software engineering: The software process- client, developer and software development life cycle: user requirement phase, specification phase, Design phase, implementation phase, Integration phase, maintenance phase, improving the software process, capability maturity models, costs and benefits of software process management. Software life cycle models and comparison of all life cycle models.

UNIT - 2 (12 Hours)

Requirements gathering- Data flow modeling, behavioural modeling, Data dictionary, data flow diagrams. IEEE standards for software requirements. Effort estimation and scheduling, Cost estimation models- Function point analysis and COCOMO . Basic design concepts: Cohesion and its various types, coupling and its various types and partitioning.

UNIT - 3 (12 Hours)

Object modeling using UML: UML overview, nature and purpose of models. UML concepts – UML view, static view, use case view, state machine view, activity view, interaction view.
Sample Tool- Argo UML, an open source tool.

UNIT - 4 (12 Hours)

Managing software project, project planning, process planning- the standard process, requirement change management, quality management, Risk management, the project management plan team structure, communication, scheduling, quality planning, measurement and tracking planning, team programming aspects, software configuration management. Project execution, project monitoring and control, project closure performing, closure analysis, closure analysis report.

Recommended Readings:

1. Stephen R.Schah ; Object Oriented and Classical Software Engineering; TMH.
2. James Rumbaugh, Ivar Jacobson, Grady Booch; The Unified Modeling Language Reference Manual, Pearson education; 2/e.
3. Pankaj Jalote ; Software Project Management in practice; PEA.
4. Roger S. Pressman ; Software Engineering – A practitioner's approach; McGraw Hill; 6/e.
5. K. K. Aggarwal and Yogesh Singh ; Software Engineering ; New Age Publications.
6. J.Rumbaugh et al; Object Oriented Modelling & Design; PHI.
7. Argo UML – www.argouml.org

List of Experiments:

Implement the database application and document the various phases of its development. The implementation can be standalone or web based. A report must be submitted that contains the following.

1. Introduction of the project (Project selection and planning).
2. Create Software Requirement Specification (SRS) document as per IEEE.
3. To study different types of Life Cycle Models and use the appropriate model for the project.

4. To draw Functional Description (DFD).
5. To draw Behavioral Description (Use case diagram) using Argo UML.
6. To draw class diagram using ArgoUML.
7. To draw state diagram using ArgoUML.
8. To draw activity diagram using ArgoUML.
9. To draw sequence diagram using ArgoUML.
10. Project Implementation.
11. Conclusion.

IT 4.5 DESIGN AND ANALYSIS OF ALGORITHMS

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
IT 4.5	Design and Analysis of Algorithms	3	1	2	3	100	25	--	--	25	**150**

Course Objectives:

1. To learn techniques for the design and analysis of efficient algorithms.
2. To understand common algorithms, algorithmic paradigms, and data structures used to solve various problems.
3. Understand the difference between the lower and upper bounds of various problems and their importance in deciding the optimality of an algorithm.

Course Outcomes:

The student after undergoing this course will be able to:

1. To demonstrate how the worst-case time complexity of an algorithm is defined.
2. Compare the efficiency of algorithms using asymptotic complexity.
3. Design efficient algorithms using standard algorithm design techniques.

UNIT - 1 (12 Hours)

Algorithm Analysis & Complexity:
Algorithm Definition and Specification, Performance analysis (Space complexity, Time complexity, Asymptotic Notations), Recurrences (methods), Performance measurement, Performance analysis of recursive algorithms, Recursion, Towers of Hanoi problem, Comparison of recursion and Iteration, Dynamic Storage Management, Garbage Collection.

UNIT - 2 (12 Hours)

Divide and Conquer strategy: General method, Binary search, Finding Maximum and Minimum, Merge sort technique, Quick sort technique
Greedy method strategy: General method, Knapsack problem, Job sequencing with deadlines, Minimum cost Spanning trees(Prims & Kruskals algorithm), Optimal storage on tapes, Optimal merge patterns, Single source Shortest paths.

UNIT - 3 (12 Hours)

Dynamic Programming: General method, Multistage graphs, All pairs shortest paths, Single Source Shortest paths, Knapsack problem, Travelling Sales person problem, Flow Shop Scheduling. Search & Traversal Techniques: Techniques for graphs- Breadth first search, Depth first search, D search, Connected components and spanning trees, Biconnected components, Code Optimization. Text processing algorithms (pattern matching).

UNIT - 4 (12 Hours)

Backtracking: General method, Sum of subsets Problem, Graph Coloring, Hamiltonian Cycles.
NP-Hard and NP-Complete Problems: Basic concepts- non-deterministic algorithms, NP-Hard and NP-Complete classes, COOK's theorem, NP-Hard Scheduling Problems, NP-Hard Code generation Problems.

Recommended Readings:

1. E.Horowitz and S. Sahini ; Fundamentals of Computer Algorithms; Galgotia publication.
2. T.H.Cormen, C.E. Leiserson, R.L.Rivest ; Introduction to Algorithms; PHI.
3. Aho Hopcraft and Ulman ; The Design and Analysis of Computer Algorithms; Addison Wesley.
4. Brassord and Bratley ; Fundamentals of Algorithms; PHI.
5. Robert Sedjewick; Algorithms; Addison Wesley.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. To implement the following using array as data structure and analyze its time complexity
   a. Insertion sort .
   b. Selection sort .
   c. Bubble sort .
   d. Quick sort .
   e. Merge sort .
2. To implement Linear and Binary search and analyze its time complexity.
3. To implement Optimal Binary Search Tree problem and analyze its time complexity.
4. To implement Dijkstra's algorithm and analyze its time complexity.
5. To implement minimum spanning trees using Kruskal's algorithm.
6. To implement minimum spanning trees using Prim's algorithm.
7. To implement a program for travelling salesman problem.
8. To implement DFS and BFS and analyze their time complexities.
9. To implement following string matching algorithms and analyze time complexities:
   a) Rabin Karp
   b) Knuth Morris Pratt
10. To implement Hamiltonian cycle problem

IT 4.6 MICROPROCESSOR AND INTERFACING

Subject Code	Name of the Subject	Scheme of Instruction Hrs/Week			Scheme of Examination						
		L	T	P	Th Duration (Hrs)	Marks					
						Th	S	TW	P	O	Total
IT 4.6	Microprocessor and Interfacing	3	1	2	3	100	25	--	25	--	**150**

Course Objectives:

The subject aims to provide the student with:

1. An in-depth understanding of the Intel 8086 architecture and programming model.
2. An ability to write Assembly language programs for a given task.
3. An understanding of different types of memories, peripheral IC's like 8255, 8259 and 8254and their interfacing with the processor.
4. An ability to interface various I/O devices with the processor.

Course Outcomes:

The student after undergoing this course will be able to:

1. Describe the architecture and explain the working of each block in 8086 processor.
2. Analyze the instruction set of 8086 processor.
3. Analyze the timing sequence of various instructions.
4. Create Assembly language programs for a given task.
5. Explain the basic programmable ICs like 8255, 8259 and 8254.
6. Design interfacing of memories and various I/O devices with the processor.

UNIT - 1 (12 Hours)

Microprocessor 8086: Detail study of 8086 architecture, addressing modes, instruction formats, data transfer instructions, string instructions, logical instructions, arithmetic instructions, processor control instructions, comparison of 8086 with 8088, assembly language programming, assembly process, assembler directives, procedures-far procedures, near procedures, parameter passing techniques, macros, macro advantages.

UNIT - 2 (12 Hours)

8086 CPU Module: Basic 8086 CPU design, generating system clock and reset signals, microcomputer bus type and buffering techniques. System Bus Structure:

Basic 8086 configurations, maximum and minimum mode, system bus timing, interrupts and interrupt responses.

8087 Coprocessor: Architecture, connection and cooperation with main processor, Instruction Set of 8087, Programming with the Arithmetic Coprocessor. Use of floating point ADD/SUB/MUL/DIV instructions, Use of F.P. instruction for generating Sine/Cosine/Exp/Log functions.

UNIT - 3 (12 Hours)

Interfacing: Programmable Peripheral Interface (PPI): Basic Description of 8255, Architecture, Modes of operation, programming the 8255.
Programmable timer8253/8254,
Interrupt Controller: Features of 8259, block diagram of 8259, Interrupt sequence, priority modes and other features Programming the 8259 and interfacing.
Brief introduction to DMA controller and keyboard, Video controller.
System Design of 8086 using Memory chips and simple I/O devices using interfaces.

UNIT - 4 (12 Hours)

80386 Architecture : Architecture and signal descriptors, Register organization, Addressing modes, Extended instruction set.
Real mode operation of 80836: Real mode operation, Memory addressing and interrupt processing.
Protected mode operation of 80386: Protected mode operation, memory organization – segmentation, descriptor types, and paging, interrupt processing in protected mode.
80386 Memory Management Unit: MMU, virtual memory, descriptor tables GDT, LDT, IDT
Review processors from 80486 onwards.

Recommended Readings:

1. John F. Uffenbeck ; The 8086/8088 family design, programming and interfacing; PHI.
2. Douglas V. Hall; Microprocessors and Interfacing: Programming and Hardware; TMH.
3.Liu and Gibson ; Microprocessor Systems: The 8086/8088 family architecture programming and design; PHI.
4. Gaonkar; Microprocessor Architecture, Programming and Applications; PHI.
5. M. Rafiquzzaman; Microprocessor and Microcomputer Based Systems; PHI.

List of Experiments:

(At least 8 experiments should be conducted from the list of experiments.)

1. Assembly language programming for 8086.

   Study of instruction set, Use of MUL/DIV instructions, Use of string processing instruction, use of XLAT instruction for code conversion.
2. Assembly language programming for 8086/8087

   Study of NDP instruction set, Use of floating point ADD/SUB/MUL/DIV instructions, Use of F.P. instruction for generating Sine/Cosine/Exp/Log functions.
3. Use of ROM-BIOS services.
4. Uses of DOS interrupt services.
5. Programs based on 386 addressing modes.
6. Programs based on bit manipulation instructions using assembly language or C.
7. Programs to find square-root of 16-bit number.
8. Interfacing keyboard.
9. Interfacing display controller.
10. Elevator problem.