With effect from Academic Year 2014-15

SCHEME OF INSTRUCTION & EXAMINATION
B.E. II YEAR
INFORMATION TECHNOLOGY

Semester - I

S.No	Syllabus Ref.No	Subject	Scheme of Instruction		Scheme of Examination			Credits
			Periods per Week		Duration in Hrs.	Maximum Marks		
			L/T	D/P		Semester-End Examination	Sessional	
		THEORY						
1	IT 211	Discrete Structures	4	-	3	75	25	3
2	IT 212	Data Structures	4/1	-	3	75	25	3
3	IT 213	Digital Electronics & Logic Design	4	-	3	75	25	3
4	IT 214	Data Communications	4	-	3	75	25	3
5	EC 215	Basic Electronics	4	-	3	75	25	3
6	MB 214	Managerial Economics & Accountancy	4	-	3	75	25	3
		PRACTICAL						
7	IT 215	Data Structures Lab	-	3	3	50	25	2
8	EC 218	Basic Electronics Lab	-	3	3	50	25	2
9	IT 216	Mini Project - I	-	3	3	-	25	1
		TOTAL	24/1	9	-	550	225	23

With effect from Academic Year 2014-15

SCHEME OF INSTRUCTION & EXAMINATION
B.E. II YEAR
INFORMATION TECHNOLOGY

Semester - II

S.No	Syllabus Ref.No	Subject	Scheme of Instruction		Scheme of Examination			Credits
			Periods per Week		Duration in Hrs.	Maximum Marks		
			L/T	D/P		Semester-End Examination	Sessional	
		THEORY						
1	MT 222	Probability and Random Processes	4	-	3	75	25	3
2	IT 221	Java Programming	4	-	3	75	25	3
3	IT 222	Design and Analysis of Algorithms	4/1	-	3	75	25	3
4	IT 223	Theory of Automata	4/1	-	3	75	25	3
5	IT 224	Software Engineering	4	-	3	75	25	3
6	IT 225	Computer Organization &Microprocessors	4	-	3	75	25	3
		PRACTICAL						
7	IT 226	Java Programming &Algorithms Lab	-	3	3	50	25	2
8	IT 227	Microprocessors Lab	-	3	3	50	25	2
9	IT 228	Mini Project - II	-	3	3	-	25	1
		TOTAL	24/2	9	-	550	225	23

With effect from Academic Year 2014-15

IT 211

DISCRETE STRUCTURES

Instruction	4 periods per week
Duration of Semester-End Examination	3Hours
Semester-End Examination	75Marks
Sessional	25Marks
Credits	3

Course Objectives:

1. Learn mathematical concepts like sets, functions, logic and be able to apply them in solving logic oriented problems and introduce useful abstractions in problem solutions and representations that have application in many areas of computer science
2. Students will be able to use graphs to model relationships, analyse data, apply probability concepts and use recursive functions and solve problems.
3. Further develop the mathematical concepts and technique which should serve as a preparation for more advanced quantitative courses.

Course Outcomes:

Upon successful completion of this course

1. Students get acquainted with the precise vocabulary and powerful notation used in formal computer science study
2. Improved thinking skills will enhance the quality of work in area of computer science.
3. Students will be able to solve complex problems using logic.

Prerequisites:

1. Elementary Algebra.
2. Introductory computer science course with C and C++

UNIT – I

Logic – Sets and Functions: Logic, Propositional equivalences – Predicates and Quantifiers – Nested Quantifiers-Rules of Inference-Sets-Set Operations, Functions.

Integers: The Integers and Division, Integers and Algorithms, Applications of Number Theory.

UNIT – II

Mathematical Reasoning, Induction, and Recursion: Proof Strategy, Sequence and Summation, Mathematical Induction, Recursive Definitions and Structural Induction, Recursive Algorithms.

Counting: Basics of Counting, Pigeonhole Principle, Permutations and Combinations– Binomial Coefficients, Generalized Permutations and Combinations, Generating Permutations and Combinations.

UNIT – III

Advanced Counting Techniques: Recurrence Relations, Solving Linear Recurrence Relations, Divide and Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion–Exclusion, Applications of Inclusion – Exclusion.

UNIT – IV

Relations: Relations & their Properties, N-ary Relations and Applications, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings.

Graphs: Graphs and Graph Models, Graph Terminology, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest Path Problems, Planar Graphs, Graph Coloring.

UNIT –V

Trees: Introduction to Trees, Application of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees.

Boolean Algebra: Boolean function, Representing Boolean functions, Logic Gates, Minimization of Circuits.

Textbook

1. Kenneth H Rosen, "Discrete Mathematics and its applications", Sixth Edition, McGraw Hill, 2006.

Suggested Reading

1. J. K. Sharma, "Discrete Mathematics", Second edition, Macmillan, 2005.
2. J.P.Trembly, R.Manohar, "Discrete Mathematical Structure with Application to Computer Science", McGraw- Hill – 1997.
3. Joel. Mott. Abraham Kandel, T.P.Baker, "Discrete Mathematics for Computer Scientist &Mathematicans", Prentice Hail N.J.,
4. C.L. Liu, "Elements of Discrete mathematics", McGraw-Hill, Third Edition.
5. U.S. Gupta, "Discrete Mathematical Structures", Pearson, 2014.

With effect from Academic Year 2014-15

IT 212

DATA STRUCTURES

Instruction	4 periods per week
Tutorial	1 period per week
Duration of Semester- End Examination	3 Hours
Semester- End Examination	75 Marks
Sessional	25 Marks
Credits	3

Course Objectives:

1. To develop proficiency in the specification, representation, and implementation of abstract data types and data structures.
2. To get a good understanding of applications of data structures.
3. To develop a base for advanced computer science study.

Course Outcomes:

Upon successful completion of this course, student will be able to

1. Choose the data structures that effectively model the information in a problem.
2. Design, implement, test, and debug programs using a variety of data structures including hash tables, binary and general tree structures, search trees, heaps, graphs, and B-trees.
3. Assess how the choice of data structures and algorithm design methods impacts the performance.

Prerequisites:

Good programming knowledge in C & CPP.

UNIT- I

Arrays, Linked Lists, and Recursion: Using Arrays, Storing Game Entries in an Array, Sorting an Array, Two-Dimensional Arrays, Singly Linked Lists, Implementing a Singly Linked List, Insertion to the Front of a Singly Linked List, Removal from the Front of a Singly Linked List, Implementing a Generic Singly Linked List, Doubly Linked Lists, Insertion into a Doubly Linked List, Removal from a Doubly Linked List, Circularly Linked Lists, Reversing a Linked List, Recursion, Linear Recursion, Binary Recursion, Multiple Recursion, Analysis of Algorithms.

UNIT- II

Stacks, Queues, and Deques: Stacks, The Stack Abstract Data Type, The STL Stack, A C++ Stack Interface, A Simple Array-Based Stack Implementation, Implementing a Stack with a Generic Linked List, Reversing a Vector Using a Stack, Matching Parentheses and HTML Tags, Queues, The Queue Abstract Data Type, The STL Queue, A C++ Queue Interface, A Simple Array-Based Implementation, Implementing a Queue with a Circularly Linked List, Double-Ended Queues, The Deque Abstract Data Type, The STL Deque, Implementing a Deque with a Doubly Linked List.

List and Iterator ADTs: Lists, Node-Based Operations and Iterators, The List Abstract Data Type, STL Lists, STL Containers and Iterators.

UNIT- III

Trees: General Trees, Tree Definitions and Properties, Tree Functions, A C++ Tree Interface, A Linked Structure for General Trees, Tree Traversal Algorithms, Depth and Height, Preorder Traversal, Postorder Traversal, Binary Trees, The Binary Tree ADT,A C++ Binary Tree Interface, Properties of Binary Trees, A Linked Structure for Binary Trees, A Vector-Based Structure for Binary Trees, Traversals of a Binary Tree, Representing General Trees with Binary Trees.

Heaps and Priority Queues: The Priority Queue Abstract Data Type, Keys, Priorities, and Total Order Relations, Comparators, The Priority Queue ADT, A C++ Priority Queue Interface, Sorting with a Priority Queue, The STL priority queue Class, Implementing a Priority Queue with a List, Selection-Sort and Insertion-Sort, Heaps, The Heap Data Structure, Complete Binary Trees and Their Representation, Implementing a Priority Queue with a Heap, Bottom-Up Heap Construction.

UNIT- IV

Search Trees: Binary Search Trees, Searching, Update Operations, C++ Implementation of a Binary Search Tree, AVL Trees, Update Operations, Splay Trees, Splaying, When to Splay, Amortized Analysis of Splaying, Tree, Multi-Way Search Trees, Update Operations for (2,4) Tree, Red-Black Trees, Update Operations.

Graph Algorithms: Graphs, The Graph ADT, Data Structures for Graphs, The Edge List Structure, The Adjacency List Structure, The Adjacency Matrix Structure, Graph Traversals, Depth-First Search, Implementing Depth-First Search, Breadth-First Search, Directed Graphs, Traversing a Digraph, Transitive Closure, Directed Acyclic Graphs, Shortest Paths,

Weighted Graphs, Dijkstra's Algorithm, Minimum Spanning Trees, Kruskal's Algorithm, The Prim-Jarn´ık Algorithm.

UNIT- V

Hash Tables: Hash Tables, Bucket Arrays, Hash Functions, Hash Codes, Compression functions, Collision-Handling Schemes, Load Factors and Rehashing.

Sorting: Merge-Sort , Divide-and-Conquer, Merging Arrays and Lists, The Running Time of Merge-Sort , Merge-Sort and Recurrence Equations, Quick-Sort, Randomized Quick-Sort, Studying Sorting through an Algorithmic Lens, A Lower Bound for Sorting, Linear-Time Sorting: Bucket-Sort and Radix-Sort, Comparing Sorting Algorithms.

Strings: Pattern Matching Algorithms, Brute Force, The Boyer-Moore Algorithm, The Knuth-Morris-Pratt Algorithm, Tries, Standard Tries, Compressed Tries, Suffix Tries.

Text Book:

1. Michael T. Goodrich, Roberto Tamassia, David M. Mount, "Data Structure and Algorithms in C++", 2nd Edition, John Wiley, 2011.

Suggested Reading:

1. Ellis Horowitz, Dinesh Mehta, S. Sahani, "Fundamentals of Data Structures in C++", Universities Press, 2007.
2. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", 3rd edition Addison-Wesley, 2007.
3. Bruno R. Preiss, "Data Structures and Algorithms with Object-Oriented Design Patterns in C++", John Wiley & Sons, 2001.
4. D. Samantha, "Classic Data Structures", Prentice Hall India, 2nd Edition, 2013.
5. Langsam, Augenstein&Tenenbaum,"Data Structures Using C & C++", 2nd edition, Prentice Hall

With effect from Academic Year 2014-15

IT 213

DIGITAL ELECTRONICS & LOGIC DESIGN

Instruction	4 periods per week
Duration of Semester- End Examination	3 Hours
Semester- End Examination	75 Marks
Sessional	25 Marks
Credits	3

Course Objectives:

1. To familiarize the students with the principles of digital Hardware.
2. To explain the operation and design of combinational and arithmetic logic circuits.
3. To facilitate with the concepts of HDL tools.

Course Outcome:

After taking the course, the students will be able to

1. Design complex logic circuits, do simplification, analysis and synthesis.
2. Understand the principles of different combinational and arithmetic logic designs.
3. Know the working principles of Latches, Flip-flops, and counters.

Prerequisites:

Physics and Mathematics.

UNIT – I

Design Concepts – Digital Hardware, Design process, Design of digital hardware Introduction to logic circuits – Variables and functions, Logic gates and networks. Boolean algebra, Synthesis using AND, OR, and NOT Gates, Design examples. Optimized implementation of logic functions – Karnaugh Map, Strategies for minimization, minimizing Product-of-Sum Forms, Incompletely Specified functions, multiple output circuits. NAND and NOR logic networks, Introduction to CAD tools and Very High Speed Integrated Circuit Hardware Description Language (VHDL).

UNIT – II

Programmable logic devices: general structure of a Programmable Logic Array (PLA), gate level diagram, schematic diagram, Programmable Array Logic (PAL) Structure of CPLDs and FPGAs, 2-input and 3-input lookup tables (LUT). Design of Arithmetic-circuits, VHDL for Arithmetic-circuits Combinational circuit building blocks – Multiplexers, Decoders, Encoders, Code converters, Arithmetic comparison circuits. VHDL for Combinational circuits.

UNIT – III

Basic Latch Gated SR Latch, Gated D Latch, Master-Slave and Edge- Triggered D Flip-Flops, D Flip-Flops with Clear and Preset. T Flip-flop, JK Flip-flop, Excitation tables. Registers-Shift Register, Parallel-Access Shift Register, Counters-Asynchronous and synchronous counters, BCD counter, Ring counter, Johnson counter, Registers and Counters in VHDL Code.

UNIT – IV

Synchronous Sequential Circuits – Basic design steps.State-Assignment problem Moore and Mealy state model.Design of Finite state machines using VHDL.State minimization, FSM as an Arbiter Circuit, Analysis of Synchronous sequential Circuits. Algorithmic State Machine (ASM) charts, formal model.

UNIT – V

Asynchronous Sequential Circuits – Behaviour, Analysis, Synthesis, State reduction, State Assignment, examples. Hazards: static and dynamic hazards. Significance of Hazards. Clock skew, set up and hold time of a flip-flop, Shift and add multiplier, data path circuit for the multiplier, ASM chart and data path circuit for the divider control circuit, sort operation.

Text book:

1. Stephen Brown, ZvonkoVranesic, "Fundamentals of Digital Logic with VHDL design", 2nd Edition, McGraw Hill, 2009.

Suggested Reading:

1. Jain R.P., "Modern Digital Electronics," 3rd edition, TMH, 2003.
2. John F. Wakerly, "Digital design Principles & Practices", 3rd edition, Prentice Hall, 2001
3. M. Morris Mano, Charles R. Kime, "Logic and Computer Design Fundamentals", 2nd edition, Pearson Education Asia, 2001.
4. ZVI Kohavi, Switching and Finite Automata Theory, 2nd edition, Tata McGraw Hill, 1995.
5. William I Fletcher, "An Engineering Approach to Digital Design", Eastern Economy Edition, PHI
6. H.T. Nagle, "Introduction to Computer Logic", Prentice Hall, 1975.

With effect from Academic Year 2014-15

IT 214

DATA COMMUNICATIONS

Instruction	4 periods per week
Duration of Semester- End Examination	3 Hours
Semester- End Examination	75 Marks
Sessional	25 Marks
Credits	3

Course Objectives:

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

1. Familiarize with the basics of data transmission, transmission media, data Communications System and its components.
2. Describe various encoding and modulation schemes, various data link protocols for flow control, error detection and correction.
3. Understand different types of multiplexing and spread spectrum techniques, Familiarize with different types of Ethernet and to understand the architecture and services of WLANs and Bluetooth.

Course Outcomes:

After completing this course the student should acquire the knowledge and ability to:

1. Demonstrate systematic understanding of Data Communication Techniques and solve problems related to data communications.
2. Apply appropriate Analytical Techniques to critically evaluate communication processes.
3. Familiarize with the basic protocols of data link layer and prepared to take the computer networks course.

Prerequisites:

Engineering physics

UNIT-I

Data Communications, Data Networks and The Internet: Data Communications and Networking for Today's Enterprise, Communications Model, Data Communications, Networks, The Internet, An Example Configuration

The Need for a Protocol Architecture, The TCP/IP Protocol Architecture, The OSI Model, Standardization within a Protocol.

Data Transmission: Concepts and Terminology, Analog and Digital Transmission, Transmission Impairments, Transmission media.

UNIT-II

Data Encoding: Digital Data Digital Signals, Digital Data-Analog Signals, Analog Data-Digital Signals, Analog Data-Analog Signals.

Data Communication Interface: Asynchronous and Synchronous Transmission, Line Configuration, Interfacing.

Data Link Control: Flow Control, Error Detection, Error Control, HDLC, Other Data link Control Protocols, Performance Issues**.**

UNIT - III

Multiplexing: Frequency Division Multiplexing, Wavelength Division Multiplexing, Synchronous TimeDivision Multiplexing, Statistical TimeDivision Multiplexing. Asymmetric Digital Subscriber Line, xDSL.

Spread Spectrum: The Concept, Direct Sequence Spread Spectrum, Frequency Hopping Spread Spectrum, Code Division Multiple Access.

UNIT -IV

Circuit Switching and Packet Switching: Switched Communications Networks, Circuit-Switching Networks,Circuit-Switching Concepts, Soft switch Architecture,Packet-Switching Principles,X.25,Frame Relay.

ATM : Architecture, Logical Connection, ATM Cells, Transmission of ATM cells.

UNIT -V

Traditional Ethernet: Topologies and Transmission Media, LAN protocol architecture, MAC sub layer, CSMA/CD, Physical Layer, Bridged, Switched and Full Duplex Ethernets

Fast Ethernet: MAC sub Layer, Physical layer, Gigabit Ethernet: MAC sub Layer, Physical Layer

Wireless LANs: Overview, Wireless LAN Technology, IEEE 802.11 Architecture and Services, IEEE 802.11 Medium Access Control, IEEE 802.11 Physical Layer.

Bluetooth: Architecture, Layers.

Text Books:

1. Behrouz A. Forouzan, "Data Communications and Networking", 4th edition, Tata McGraw Hill, 2006.
2. William Stallings, "Data and Computer communication", 8th edition, Pearson Education, Asia-2004.

Suggested Reading:

1. Fred Halsall, "Data Communications, Computer Networks and Open Systems", 4th edition, Pearson Education, 2000.
2. Andrew S. Tanenbaum, "Computer Networks", 5th edition, Pearson Education.
3. Gilbert Held, "Understanding Data Communications", 7th Edition, Pearson Education.
4. Nader F.Mir, "Computer and Communication Networks, 7th edition, Prentice Hall,

With effect from Academic Year 2014-15

EC 215

BASIC ELECTRONICS
(Common for CSE, IT, MECH, PROD)

Instruction	4 Periods per week
Duration of University Examination	3 Hours
University Examination	75 Marks
Sessionals	25 Marks
Credits	3

Course Objectives:

1. To understand the knowledge of basic semiconductor devices and create foundation for forthcoming circuit design courses
2. To understand various applications like amplifiers, oscillators and op-amps also motivate and train students in logic design.
3. To understand the working principle of the transducers and aware the students about the advances in Instrumentation.

Course Outcomes:

1. Ability to understand the usefulness of semiconductor devices in circuit making like rectifiers, filters, regulators etc.
2. Ability to develop new directions in logic design to analyze, design and implement combinational circuits.
3. Ability to analyze the principles and practices for instrument design to development the real world Problems.

UNIT – I

Semiconductor Theory: Energy levels, Intrinsic and Extrinsic Semiconductor, Mobility, Diffusion and Drift current, Hall effect, Law of mass action, Characteristics of P-N Junction diode, current equation, Parameters and Applications.

Rectifiers: Half wave and Full wave Rectifiers Bridge and center tapped with and without filters, Ripple factor, regulation and efficiency.

UNIT – II

Transistors: Bipolar and field effect transistors with their h-parameter equivalent circuits, Basic Amplifiers classification and their circuits (Qualitative treatment only).

Regulators and Inverters: Zener Diode**,** Breakdown mechanisms, Characteristics, Effect of Temperature, Application as voltage regulator.

UNIT-III

Feedback Amplifiers: Properties of Negative Feedback Amplifier, Types of Negative Feedback, Effect of negative feedback on Input impedance and Output impedance, Applications (Qualitative treatment only).

Oscillators: principle of oscillations, LC Type-Hartley, Colpitt and RC Type- Phase shift, Wien Bridge and Crystal Oscillator (Qualitative treatment only).

UNIT – IV

Operational Amplifiers: Basic Principle, Ideal and practical Characteristics and Applications-Summer, Integrator, Differentiator, Instrumentation Amplifier.

Digital System: Review of basic gates, Universal gates, Demorgan's theorem, minimization with Karnaugh Map up to three variables and realization of half, Full Adder and half, Full Sub tractors.

UNIT – V

Data Acquisition systems: Study of transducers-LVDT, Strain gauge.
Photo Electric Devices and Industrial Devices: Photo diode, Photo Transistor, LED, LCD, SCR, UJT Construction and Characteristics and their applications only.
Display Systems: Constructional details of C.R.O and Applications.

Text Books:

1. Robert L. Boylestad, Louis Nashelsky, "Electronic Devices and Circuits Theory", Pearson Education, 9^{TH} edition, LPE, Reprinted, 2006.
2. Morris Mano, "Digital Design", Pearson Education, Asia 2002.

Suggested Reading:

1. Jacob Millman and C., Halkias, "Electronic Devices", McGraw Hill, Eight Edition, Reprinted, 1985.
2. Ramakanth A. Gayakwad, "Op-Amps and Linear Integrated Circuits", Prentice Hall of India, 3rd edition,1985
3. W. D. Cooper, A. Helfric, "Electronic Instrumentation and Measurement Techniques", PHI, 4th edition.
4. S. Shalivahan, N. Suresh Kumar, A Vallavea Raj, "Electronic Devices and Circuits", Tata McGraw Hill, 2003

With effect from Academic Year 2014-15

MB 214

MANAGERIAL ECONOMICS AND ACCOUNTANCY

Instruction:	4L periods per week
Duration of Main Examination	3 Hours
Main Examination	75 Marks
Internal Examination	20 Marks
Case Study/ Assignment	5 Marks
Credits	3

Objective: The objective of the course is to provide the analytical tools and managerial insights that are essential for the solution of those business problems that have significant consequences for the firm and society.

UNIT-I: Introduction to Managerial Economics
Introduction to Economics and its evolution - Managerial Economics - its scope, importance, Its usefulness to engineers - Basic concepts of Managerial economics.

UNIT-II:Demands Analysis
Demands Analysis - Concept of demand, determinants, Law of demand, its assumptions, Elasticity of demand, price, income and cross elasticity, Demand Forecasting - Markets Competitive structures, price-output determination under perfect competition and Monopoly. (Theory questions and small numerical problems can be asked).

UNIT-III: Production and Cost Analysis
Theory of Production - Firm and Industry - Production function - input-out relations - laws of returns - internal and external economies of scale. Cost Analysis: Cost concepts - fixed and variable costs - explicit and implicit costs - out of pocket costs and imputed costs - Opportunity cost - Cost output relationship - Break-even analysis. (Theory and problems).

UNIT-IV:Capital Management
Capital Management, its significance, determinants and estimation of fixed and working capital requirements, sources of capital - Introduction to capital budgeting, methods of payback and discounted cash flow methods with problems.
(Theory questions are numerical problems on estimating working capital requirements and evaluation of capital budgeting opportunities can be asked).

UNIT-V: Accountancy
Book-keeping, principles and significance of double entry book keeping, Journal, Subsidiary books, Ledger accounts, Trial Balance concept and preparation of Final Accounts with simple adjustments.
(Theory questions and numerical problems on preparation of final accounts, cash book, petty cash book, bank reconciliation statement).

Essential Readings:

1. Mehta P.L., “Managerial Economics – Analysis, Problems and Cases”, Sulthan Chand & Son’s Educational publishers, 2011.
2. Maheswari S.N. “Introduction to Accountancy”, Vikas Publishing House, 2005.
3. Panday I.M. “Financial Management”, Vikas Publishing House, 2009.

Suggested Readings:

1. Varshney and KL Maheswari, Managerial Economics, Sultan Chand, 2001.
2. M.Kasi Reddy and S.Saraswathi, Managerial Economics and Financial Accounting, Prentice Hall of India Pvt Ltd, 2007.
3. JC Pappas and EF Brigham, Managerial Economics.

With effect from Academic Year 2014-15

IT 215

DATA STRUCTURES LAB

Instruction	4 periods per week
Duration of Semester- End Examination	3 Hours
Semester- End Examination	75 Marks
Sessional	25 Marks
Credits	3

Course Objectives:

1. The fundamental design, analysis, and implementation of basic data structures and algorithms.
2. Understand data structures such as Trees, Threaded Binary Trees, Heaps, etc.
3. Be familiar with graph operations and algorithms.
4. Get familiar with advanced tree structures like AVL, Splay, m-way, B-Trees.

Course Outcomes:

Student will be able to

1. Choose the data structures that effectively model the data in a problem.
2. Design, implement, test, and debug programs using a variety of data structures including hash tables, binary and general tree structures, heaps, graphs, and B-trees.
3. Choose the appropriate data structure and algorithm design method for a specified application.

Prerequisites:

Good programming knowledge in C & CPP.

List of Programs

1. Implement String ADT.
2. Implement Infix to Postfix Conversion and evaluation of postfix expression using Stack.
3. Implementation of Queues, Circular Queues and Deques.
4. Implementation of Single Linked List and its operations.
5. Implementation of Double Linked List and its operations.
6. Implementation of Binary Search and Hashing.
7. Implementation of Sorting Techniques: Insertion, Bubble, Selection, Shell, Merge, Quick, Heap.
8. Implementation of Tree Traversals on Binary Trees.
9. Implementation of operations on AVL Trees.
10. Implementation of Traversal on Graphs.
11. Implementation of Splay Trees.

Suggested Reading:

1. Michael T. Goodrich, Roberto Tamassia, David M. Mount, "Data Structure and Algorithms in C++", 2nd Edition, John Wiley, 2011.
2. Ellis Horowitz, Dinesh Mehta, S. Sahani, "Fundamentals of Data Structures in C++", Universities Press, 2007.
3. K. R. Venugopal, "Mastering C++", Tata McGraw-Hill Publishing Company, 1997
4. D. Samantha, "Classic Data Structures", Prentice Hall India, August 2004.
5. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", 3rd edition, Addison-Wesley, 2007

With effect from Academic Year 2014-15

EC 218

BASIC ELECTRONICS LAB

(Common for CSE, IT, MECH, PROD)

Instruction	3 Periods per week
Duration of University Examination	3 Hours
University Examination	50 Marks
Sessionals	25 Marks
Credits	2

Course Objectives:

The main objectives of this course are:

1. To study the electronics components.
2. To study characteristics of semi-conductor devices.
3. To study simple electronic circuits.

Course Outcomes:

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

1. Understand the knowledge regarding electronic components and equipment.
2. Design various rectifiers and filters .Analysis of characteristic behavior of BJT , FET
3. Design of an amplifier.
4. Verify the operation of Op-amp for various applications.

List of Experiments:

1. Study of Electronic components.

2. Characteristics of Semiconductor diodes (Germanium, Silicon and Zener).

3. CRO and its Applications.

4. Half, Full wave rectifiers with and without filters.

5. Voltage Regulator using zener diode.

6. Characteristics of BJT in CE Configuration.

7. Characteristics of FET in CS Configuration.

8. Amplifier with and without feedback.

9. RC Phase shift oscillator

10. Operational Amplifier and its applications.

11. Verification of Logic gates

12. Realization of Half and Full adder

Suggested Reading:

1. Paul B. Zbar, Albert P. Malvino, Michael A. Miller, "Basic Electronics, A Text - Lab Manual", 7th Edition, TMH, 1994.

2. Paul B. Zbar," Industrial Electronics, A Text - Lab Manual", 3rd Edition.

With effect from Academic Year 2014-15

IT 216

MINI PROJECT - I

Instruction	3 periods per week
Sessional	25 Marks
Credits	1

Course Objectives:

1. To learn by doing, by taking responsibility of the end product.
2. To develop capability to analyse and solve real world problems with an emphasis on applying/integrating knowledge acquired.

Course Outcomes:

Students should be able to do the following:

1. Construct innovative solutions.
2. To work in team as well as individuals.
3. To manage time and resources.

The Students are required to implement one of the projects from project exercise given in the suggested readings of the theory subjects. During the implementation of the project, Personnel Software Process (PSP) has to be followed.

Report of the project work has to be submitted for evaluation.

With effect from Academic Year 2014-15

MT 222

PROBABILITY AND RANDOM PROCESSES

Instruction	4 periods per week
Duration of Semester-End Examination	3Hours
Semester-End Examination	75Marks
Sessional	25Marks
Credits	3

Course Objectives:

1. To induce the ability to describe a random experiment in terms of procedure, observation, and a Probability model.
2. To inculcate ability to characterize functions of random variables
3. To impart the students the methods to characterize stochastic processes with an emphasis on stationary random processes.

Course outcomes:

1. The student is expected to characterize jointly multiple discrete and continuous random variables
2. The student must be able to describe conditional and independent events and conditional random variables.
3. Learn the techniques to describe independent events and independent random variables and their sums.

UNIT-I

The meaning of Probability-Introduction, definitions-Probability and Induction-Causality versus Randomness.

The Axioms of probability: Set theory-Probability Space- Conditional Probability.

Repeated trials: combined experiments-Bernoulli's trials –Bernoulli's theorem and games of chance.

UNIT-II

The concept of Random variable: Introduction-Distribution and density functions-Specific random variables-Conditional distributions-Asymptotic Approximations for Binomial approximations.

Functions of one random variables: The random variable g(x)- The distribution of g(x)-Mean and variance-moments – Characteristic Function.

UNIT-III

Two random variables: Bivariate distributions-one function of two random variables

-Two function of two random variables-joint moments-joint characteristic functions-conditional distributions- conditional Expected Probability function.

UNIT-IV

Random Process: Definitions- Basic concepts and examples-Stationary and ergodicity-second order properties- Spectral representation Winer-Kinche
Theorem.

UNIT-V

Linear Operations: Gaussian Processes-Poisson process- Low pass and band pass noice representations.

Text Books:

1. T.Veerarajan "Probability, Statistics and Random Process", Tata Mc Graw Hill company Pvt. Ltd. Third Edition, 2010
2. P.Ramesh Babu "Probability Theory and Random Processes", Tata McGraw Hill Education Private Limited First Edition-2014
3. S. C.Gupta and V.K.Kapoor "Mathematical Statistics", Sultan Chand & Sons Tenth Edition, 2000.

With effect from Academic Year 2014-15

IT 221

JAVA PROGRAMMING

Instruction	4 Periods per week
Duration of Semester- End Examination	3 Hours
Semester-End Examination	75 Marks
Sessional	25 Marks
Credits	3

Course Objectives:

1. To understand fundamentals of object-oriented programming in Java which includes defining classes, invoking methods, using class libraries.
2. To create Java application programs using sound OOP practices such as interfaces, APIs and error exception handling.
3. Using API to solve real world problems.

Course Outcomes:

At the end of this course student will:

1. Achieve proficiency in object-oriented concepts and also learns to incorporate the same into the Java programming language.
2. Develop programming skills and implement problem-solving techniques using OOP concepts.
3. Develop the ability to solve real-world problems through software development in high-level programming language and Large APIs of Java.

Prerequisites:

1. Should have programming knowledge in high level language such as C.
2. Should have basic concepts of OOPs.

UNIT-I

Introduction To Java: Objects, Classes, Java Programs, Introduction to jdk and jre, Java Primitive Types, Basic Operators, Conditional and Logical statements, Some Typical Differences Between C and Java.

Defining Classes: Adding Instance Fields and Methods, Constructors, Access Modifiers (Visibility Modes), Object Creation Examples, Method Overloading and Constructor Overloading, Use of static and final keywords, Objects as parameters, Difference between local variable and instance field, Introduction to Object class, How to read user input (from keyboard).

UNIT-II

Arrays, Strings in Java: How to create and define arrays, Introduction to java.util.Array class, Difference between String &StringBuffer classes, StringTokenizer class and Wrapper classes and conversion between Objects and primitives

Inheritance, Interfaces and Packages in Java: Defining super / sub classes, Abstract classes, Method overriding, Interfaces, Using Library Interfaces [Comparable and Comparator], Creating and Defining Packages;

Inner classes in Java: Types of inner classes, Creating static / non-static inner classes, Local and anonymous inner classes.

UNIT-III

Exception Handling in Java: What are exceptions, writing your own exception classes, [try, catch, throw, throws clauses] , Difference between checked Vs.unchecked Exceptions, Error Vs. Exception.

Multithreading in Java: what are threads, how to create threads, Thread class in java, use of synchronized keyword, how to avoid deadlock.

Generics and Frameworks: Generics, Collections Framework, Collection interfaces and classes ArrayList, LinkedList, Vector.

UNIT-IV

GUI Design & Event Handling: Component, Container, Color , GUI Controls, Layout Managers, Introduction to Swings, Events, Listeners, Icon interface, Writing GUI Based applications, Applets , Running Applets.

UNIT-V

File Handling: Stream classes, Reader and Writer classes, File and Directory class

Database Handling in Java: Java Database Connectivity (JDBC)

Text Book:

1. Herbert Schildt: "Java™: The Complete Reference Java", Eighth Edition, Tata McGraw Hill Publications, 2011, ISBN: 9781259002465

Suggested Reading:

1. Cay S. Horstmann, Gary Cornell: "Core Java, Volume I--Fundamentals", 8th edition, Prentice Hall, 2008, ISBN: 9780132354790
2. K. Arnold and J. Gosling, "The JAVA programming language", 3rd edition, Pearson Education, 2000.
3. Timothy Budd, "Understanding Object-oriented programming with Java", Addison-Wesley, 2002.
4. C. Thomas Wu, "An introduction to Object-oriented programming with Java", 4th edition, Tata McGraw-Hill Publishing company Ltd., 2006.
5. Deitel&Deitel, "Java: How to Program", 9th Edition, PHI, 2014

With effect from Academic Year 2014-15

IT 222

DESIGN AND ANALYSIS OF ALGORITHMS

Instruction	4 periods per week
Tutorial	1 period per week
Duration of Semester-End Examination	3 Hours
Semester-End Examination	75 Marks
Sessional	25 Marks
Credits	3

Course Objectives:

1. To develop the problem solving capability using different algorithmic design techniques.
2. To learn how to analyse the asymptotic performance of algorithms and prove their correctness.
3. To introduce the notions of NP-completeness and NP-hardness.

Course Outcomes:

Students should be able to do the following:

1. Synthesize/adapt an algorithm to solve the problem in hand and argue its correctness.
2. Analyse best-, average- and worst-case complexities of algorithms using asymptotic notations.
3. Identify the complexity classes such as P, NP, NP-Complete and NP-Hard to which an algorithm belongs and design a feasible solution.

Prerequisites:

Programming language, Data Structures, Discrete mathematics, Basic probability theory.

UNIT-I

Introduction: Algorithm Specification, Performance analysis, Space Complexity, Time Complexity, Asymptotic Notation (O, Omega, Theta), Practical Complexities, Performance Measurement, Randomized Algorithms: An informal discussion, Review of elementary data structures : Stacks, Queues, Trees, Heap and Heap Sort, Set representation, UNION, FIND.

UNIT-II

Divide- and Conquer: The general method, Finding maximum minimum. Merge sort, Quick sort, Selection Sort, Strassen's Matrix Multiplication **Greedy Method:** The General Method,Knapsack problem, Job sequencing with deadlines, Minimum Cost Spanning Trees, Optimal Storage on tapes, Optimal merge patterns, Single Source Shortest Paths.

UNIT-III

Dynamic Programming: The General Method, Multistage graph, Single source shortest path, All Pair Shortest Path, Optimal Binary Search trees, 0/1 Knapsack, Reliability Design, Traveling Salesperson Problem, **Techniques for Graph Traversal**: Breath First Traversal, Depth First Traversal, Connected Components and Spanning Trees, Bi-connected Components and Depth First Search.

UNIT-IV

Backtracking : The General Method,8-Queens Problem, Graph Colouring, Hamilton cycle, Knapsack Problem, **Branch and Bounds:** The Method, LC Search, 15 puzzle, FIFO Branch and Bound, LC Branch and Bound, 0/1 Knapsack Problem, Traveling salesperson problem.

UNIT-V

NP-Hard and NP-Completeness: Basic concepts, Non-Deterministic Algorithms, The Classes NP Hard and NP Complete. Cook's theorem, NP-hard Graph Problems: Node Cover Decision Problem, Chromatic Number Decision Problem, Directed Hamiltonian Cycle, Traveling salesperson decision problem, NP Hard Scheduling Problems: Job Shop Scheduling.

Text Book:

1. Ellis Horowitz, SartajSahani and SanguthevarRajasekaran, Fundamentals of Computer Algorithm, 2nd edition, Semester-End Press, 2011.

Suggested Reading:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 2nd Edition, Prentice Hall of India Private Limited, 2006.
2. AnanyLevitin, "Introduction to the Design & Analysis of Algorithms", Pearson Education, 2003.
3. Aho, Hopcroft, Ullman, "The Design and Analysis of Computer Algorithm", Pearson Education, 2000.
4. ParagH.Dave, Himanshu B. Dave, "Design and Analysis of Algorithms", Pearson Education, Second Edition, 2014.

With effect from Academic Year 2014-15

IT 223

THEORY OF AUTOMATA

Instruction	4 periods per week
Tutorial	1 period per week
Duration of Semester-End Examination	3 Hours
Semester-End Examination	75 Marks
Sessional	25 Marks
Credits	3

Course Objectives:

1. To study, evaluate and explain the differences between different computational models, such as Turing machines, push-down automata, finite automata, etc.
2. To design solutions for problems using the different computational models (Pushdown Automata, Finite Automata, TMs).
3. To understand and work with grammars and representations of formal languages.

Course Outcomes:

Upon successful completion of this course, students should be able to have

1. An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
2. An ability to apply knowledge of computing and mathematics appropriate to the discipline in the modeling and design of computer-based systems in a way that demonstrates comprehension of the trade-offs involved in design choices.
3. An ability to design, implements, and evaluate a computer-based system, process, component, or program to meet desired needs.

Prerequisites: Discrete Structures and Data Structures

UNIT-I

Automata: Introduction to Finite Automata, The Central Concepts of Automata Theory: Alphabets, Strings, Languages.

Finite Automata: An Informal Picture Of Finite Automata: The Ground Rules, The Protocol, Enabling the Automata to Ignore Actions, The Entire System as an Automaton. Deterministic Finite Automata: Definition of a DFA, How a DFA Processes Strings? Simpler

Notations for DFA's, Extending the Transition Function to Strings, The Language of a DFA, Nondeterministic Finite Automata: Definition of NFA, The Extended Transition Function, The Language of an NFA, Equivalence of NFA and DFA, An Application: Text Search, Finite Automata with Epsilon-Transitions: Use of €-transitions, The formal notation for an €-NFA , €-closure, Extended Transitions and Languages for €-NFA's, Eliminating €-transitions

UNIT -II

Regular Expression and languages: Regular Expressions: The Operators of Regular Expressions, Building Regular Expressions. Finite Automata and Regular Expression: From DFAs to Regular Expressions, Converting DFA's to Regular Expressions by Eliminating States, Converting Regular Expressions to Automata. Applications of Regular Expressions, Algebraic Laws for Regular Expressions.

Properties of Regular Languages: Proving Languages not to be Regular: The pumping lemma for Regular Languages, Applications of Pumping Lemma, Closure Properties of Regular Languages: Decision Properties of Regular Languages: Testing Emptiness of Regular Languages, Testing Membership in a Regular Language. Equivalence and Minimization of Automata: Testing Equivalence of States, Testing Equivalence of Regular Languages ,Minimization of DFA's.

UNIT-III

Context Free Grammars and Languages: Context-Free Grammars: Definition of Context Free Grammars, Derivations using a Grammar, Leftmost and Rightmost Derivation, The language of a Grammar, Parse Trees: Constructing Parse Trees, The Yield of a Parse Tree, Applications of CFGs, Ambiguity in Grammars and Languages: Ambiguous Grammars, Removing Ambiguity From Grammars, Leftmost Derivations as way to Express Ambiguity, Inherent Ambiguity.

Properties of Context Free Languages: Normal Forms for Context-Free Grammars: Eliminating Useless Symbols, Computing the Generating and Reachable Symbols, Eliminating Productions, Eliminating Unit Productions, Chomsky Normal Form, Pumping Lemma for

CFL's: Statement of the Pumping Lemma, Applications of Pumping Lemma for CFL's, Closure Properties of CFL's, Decision Properties of CFL's: Testing Emptiness of CFL's, Testing Membership in a CFL's.

UNIT -IV

Pushdown Automata: Definition of pushdown automaton: The Formal Definition of PDA,

Graphical Notation for PDA's, Instantaneous Description of a PDA. The Languageof a PDA: Acceptance by Final State, Acceptance by Empty Stack, From Empty Stack to Final State, From Final State to Empty Stack, Equivalence of PDA's and CFG's: From Grammars to PDA's, From PDA's to Grammars, Deterministic Pushdown Automata: Definition, Regular Languages and Deterministic PDA's, DPDA's to CFL's, DPDA's to Ambiguous Grammars.

Introduction to Turing Machines: Problems that Computer Cannot Solve: The Turing Machine: Notation for the TM, Instantaneous Descriptions for TM's, Transitions Diagrams, The Language of a TM, Turing Machines and Halting, Programming Techniques for Turing Machines: Storage in the State, Multiple Tracks, Subroutines, Extensions to the Basic Turing Machine: Multitape Turing Machine, Equivalence of One-Tape and Multi-Tape TM's, Non deterministic Turing Machines, Restricted Turing Machines: TM's with Semi infinite Tapes, Multistack Machines, Counter Machines. Turing Machine and Computers: Simulating a Computer by a TM.

UNIT -V

Undecidability: A Language That Is Not Recursively Enumerable: Enumerating the Binary Strings, Codes for Turing Machines, The Diagonalization Language, An Undecidableproblemthat is RE: Recursive Languages, Compliments of Recursive and RE languages, The Universal Languages, Undecidablity of the Universal Language,Undecidable problems about Turing Machines: Reductions, TM's That Accept The Empty Language, Rice's Theorem and Properties of RE languages, Post's Correspondence Problem: Definition of PCP, The Modified PCP, Other UndecidableProblems.

Intractable Problems: The classes P and NP: Problems Solvable in Polynomial Time,Nondeterministic Polynomial Time, NP-Complete Problem.

Text book:

1. John E. Hopcroft, Rajeev Motwani, Jeffery D Ullman, "Introduction to AutomataTheoryLanguages and Computation", Third edition, Pearson Education, 2007.

Suggested Reading:

1. John C Martin. "Introduction to Language and Theory of Computation", 3rd edition, TMH, 2003.
2. Daniel Cohen, "Introduction to Computer Theory", 2nd edition, Wiley Publications, 2007.
3. Mishra K., Chandrasekaran N., "Theory of Computer Science (Automata, Languages and
4. Computation)", 3rd edition,Prentice Hall of India 2008.
5. ShyamalendraKandar, "Introduction to Automata Theory, Formal Languages and Computation", Pearson, 2013.

With effect from Academic Year 2014-15

IT224

SOFTWARE ENGINEERING

Instruction	4 periods per week
Duration of Semester-End Examination	3 Hours
Semester-End Examination	75 Marks
Sessional	25 Marks
Credits	3

Course Objectives:

This course introduces the students to

1. Understand the software life cycle models.
2. Understand the importance of the software development process.
3. Understand the importance of software quality and review techniques.

Course Outcomes:

1. An ability to apply knowledge of mathematics, science, and engineering.
2. An ability to design and conduct experiments, as well as to analyze and interpret data, conduct tests using various testing methods to verify and validate the results.
3. An ability to identify, formulate and implement software projects.

Pre-requisites: Knowledge of design, coding, and debugging programs.

UNIT-I

Software and Software Engineering: The Nature of Software, Software Engineering. The Software Process, Software Engineering Practice.Process Models: A Generic Process Model, Prescriptive Process Models, Specialized Process Models, The Unified Process, Process Technology, Product and Process, Process Assessment and Improvement, The CMMI, The people CMM,Introduction to Agile development.

Understanding Requirements: Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Developing Use Cases, Building the Requirements Model, Negotiating Requirements, Validating Requirements.

Requirements Modeling: Requirements Analysis, Scenario-Based Modeling, Problem Analysis, Data Flow Diagrams, Data Dictionaries, Entity-Relationship diagrams, Software Requirement and Specifications, Behavioral and non-behavioral requirements, Software Prototyping.

UNIT-II

Design Concepts: Design within the Context of Software Engineering, The Design Process, Design Concepts. Cohesion & Coupling, Classification of Cohesiveness & Coupling, Function Oriented Design, Object Oriented Design, User Interface Design.

Architectural Design: Software Architecture, Architecture Styles.

Component level Design: Designing Class-Based Components, Conducting Component-Level Design, Designing Traditional Components, Component-Based Development.

UNIT-III

Quality Concepts: Software Quality, Achieving Software Quality.

Review Techniques: Cost Impact of Software Defects.

Software Quality Assurance: Background Issues, Elements of Software Quality Assurance, SQA Tasks, Goals and Metrics, Formal Approaches to SQA, Statistical Software Quality Assurance, Software Reliability, The ISO 9000 Quality Standards, The SQA Plan.

UNIT-IV

Software Testing Strategies: A Strategic Approach to Software Testing, Strategic Issues, Validation Testing, System Testing, The Art of Debugging. Testing Tools –Rational functional tester, Win Runner, Load Runner, Testing Standards,Selenium software testing tool.

Testing Conventional Applications:

Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, Basis Path Testing, Control Structure Testing, Black–Box Testing, Software Configuration Management.

Product Metrics: A Framework for Product Metrics, Size Metrics like LOC, Token Count, Function Count, Design Metrics, Data Structure Metrics, Information Flow Metrics, Metrics for Testing, Metrics for Maintenance.

UNIT-V

Estimation: Observations on Estimation, The Project Planning Process, Software Scope and Feasibility, Resources, Software Project Estimation, Decomposition Techniques, Empirical Estimation Models, Specialized Estimation Techniques.

Risk Management: Reactive versus Proactive Risk Strategies, Software Risks, Risk Identification, Risk Projection, Risk Refinement, Risk Mitigation, Monitoring, and Management, The RMMM Plan.

Text Books:

1. Roger S.Pressman, "Software Engineering: A Practitioners Approach" ,7th edition, McGrawHill, 2009.

Suggested Reading:

1. Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, "Fundamentals of Software Engineering", PHI, 2nd edition.
2. Ali Behforoz and Frederic J.Hadson, "Software Engineering Fundamentals", Oxford Semester-End Press, 1996.
3. PankajJalote "An Integrated Approach to Software Engineering ", 3rd edition, Narosa Publishing house, 2008.
4. James F.Peters, WitoldPedrycz, "Software Engineering-An engineering Approach", John Wiley Inc., 2000.

With effect from Academic Year 2014-15

IT 225

COMPUTER ORGANIZATION & MICROPROCESSORS

Instruction	4 periods per week
Duration of Semester-End Examination	3 Hours
Semester-End Examination	75 Marks
Sessional	25 Marks
Credits	3

Course Objectives:

1. To familiarize students with the design and organization of a digital computer, operation of various functional units, instruction set design and factors that influence the performance of a computer.
2. To facilitate students with the understanding of architecture and instruction set of 8085 in particular and programming 8085.
3. To facilitate students with the understanding of the functionality and interfacing of various peripheral devices.

Course Outcomes:

After completing the course, student should be able to

1. Understand and analyze the performance of computer systems and know how to improve their efficiency.
2. Understand the instruction set of 8085 and write assembly language programs.
3. Design new special purpose systems for various applications using appropriate peripheral interfacing.

Prerequisites:

Digital Electronics and Logic Design

UNIT-I

Basic Structure of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Software, Performance, Multiprocessors and Multi-computers, Historical perspective.

Arithmetic: Addition and Subtraction of Signed numbers: Addition/Subtraction Logic Unit, Design of fast adders: Carry – Look-ahead Addition, Multiplication of positive numbers, Signed-Operand Multiplication: Booth Algorithm, Fast Multiplication: Bit-Pair Recording of Multipliers, Carry-Save addition of Summands, Integer Division, Floating Point Numbers and Operations: IEEE Standard for Floating-Point Numbers, Arithmetic Operations on Floating-Point Numbers, Guard Bits and Truncation, Implementing Floating-Point Operations.

UNIT-II

The Memory System: Basic concepts, Semi-conductor RAM Memories: Internal Organization of Memory Chips, Static Memories, Asynchronous DRAMs, Synchronous DRAMs , Structure of Larger Memories, Memory System Considerations, Rambus Memory, Read-Only Memories: ROM, PROM, EPROM, EEPROM, Flash Memory, Speed, Size and Cost, Cache Memories**:** Mapping Functions, Replacement Algorithms, Performance considerations: Interleaving, Hit rate and Miss Penalty, Caches on the Processor Chip, Other Enhancements. Virtual Memories: Address Translation, Memory Management requirements, Secondary Storage: Magnetic Hard Disks, Optical Disks and Magnetic Tape Systems.

UNIT-III

8085 Microprocessor Architecture: Introduction to Microprocessors, The 8085 MPU: The 8085 Microprocessor, Microprocessor Communication and Bus Timings, De-multiplexing the Bus AD_7-AD_0, Generating Control Signals, A Detailed Look at the 8085 MPU and its Architecture, Decoding and Executing an Instruction.

Programming the 8085: Introduction to 8085 instructions: Data Transfer (Copy) Operations, Arithmetic Operation, Logic Operations, Branch Operations, Writing Assembly Language Programs, Debugging a Program. Programming techniques with Additional instructions: Programming Techniques-Looping, Counting and Indexing, Additional Data Transfer and 16-Bit Arithmetic Instructions, Arithmetic Operations Related to memory, Logic Operations: Rotate and Compare, Dynamic Debugging.

UNIT-IV

Stacks and subroutines: Stack, Subroutine, Restart, Conditional CALL and RETURN instructions, Advanced Subroutine Concepts.

Interrupts: The 8085 Interrupt, 8085 Vectored Interrupts:TRAP, RST 7.5, 6.5, AND 5.5, Additional I/O Concepts and Processes: Programmable Interrupt Controller (8259A), Direct Memory Access (DMA).

Interfacing Data Converters: Digital to Analog (D/A) Converters, Analog to Digital (A/D) Converters.

UNIT-V

Programmable Peripheral Interface (Intel 8255A), Programmable Communication Interface (Intel 8251), Programmable Interval Timer (Intel 8253 and 8254), Programmable Keyboard /Display Controller (Intel 8279), Serial and Parallel bus Standards: RS 232 C and IEEE 488.

Text books:

1. Carl Hamacher, ZvonkoVranesic, SafwatZaky, “Computer Organization”, 5th Edition, McGraw Hill, 2002.
2. Ramesh S Gaonkar, “Microprocessor Architecture, Programming and Applications with the8085”, 5th edition, Prentice Hall, 2002.

Suggested Readings:

1. M. M. Mano, “Computer System Architecture”, 3rd edition, Prentice Hall, 1994.
2. William Stallings, “Computer Organisation and Architecture, Design for Performance”, Pearson, 9th Edition, 2013
3. Pal Chouduri, “Computer Organization and Design”, Prentice Hall of India, 1994.
4. Douglass V. Hall,“Microprocessors and Interfacing: Programming and Hardware”, 2nd Edition,

With effect from Academic Year 2014-15

IT226

JAVA PROGRAMMING AND ALGORITHMS LAB

Instruction	4 periods per week
Duration of Semester-End Examination	3 Hours
Semester-End Examination	75 Marks
Sessional	25 Marks
Credits	3

Course Objectives:

1. To build software development skills using java programming for real world applications.
2. To implement frontend and backend of an application
3. To implement classical problems using java programming.

Course Outcomes:

Students should be able to demonstrate the following.

1. Develop java based software applications using different IOs
2. Design user friendly GUI with befitting backend
3. Implement Algorithms using java programming

List of programs

1. Program(s) to illustrate the concepts polymorphism, abstract class,interface, String Handling and inner classes.
2. A program(s) for demonstrating different exceptions and creation of user defined exception.
3. A program to illustrate multithreading & thread synchronization.
4. Program(s) using Collection classes and Interfaces
5. Program(s) to illustrate the usage of filter and Buffered I/O streams.
6. An application involving GUI with different controls, event handling and applets.
7. A Program to connect to MySql database using JDBC.
8. A program to implement 0/1 Knapsack problem using Dynamic Programming.
9. A program single source shortest path using Dijkstra's algorithm.
10. A program to implement N Queen's problem using Back Tracking
11. A program to find the chromatic number of a given graph.
12. A program to obtain the Topological ordering of vertices in a given digraph.
13. A GUI based Applet Animation to implement the Travelling Salesperson problem.
14. A program to find the shortest path of the multistage graph using dynamic programming.

Suggested Reading:

1. Ellis Horowitz, SartajSahani and SanguthevarRajasekaran, “Fundamentals of Computer Algorithm”, 2nd edition, Semester-End Press, 2011.
2. Herbert Schildt: “Java™: The Complete Reference Java”, 8th edition, Tata McGraw Hill Publications, 2011.

With effect from Academic Year 2014-15

IT 227

MICROPROCESSORS LAB

Instruction	4 periods per week
Duration of Semester-End Examination	3 Hours
Semester-End Examination	75 Marks
Sessional	25 Marks
Credits	3

Course Objectives:

1. To become familiar with the architecture and Instruction set of Intel 8085 microprocessor.
2. To provide practical hands on experience with Assembly Language Programming.
3. To familiarize the students with interfacing of various peripheral devices with 8085 microprocessor.

Course Learning Outcomes:

After completing the course students should be able to

1. Describe the architecture and comprehend the instruction set of 8085.
2. Understand and apply the principles of Assembly Language Programming in developing microprocessor based applications.
3. Work with standard microprocessor interfaces like serial ports, digital-to-analog Converters and analog-to-digital converters etc.

Prerequisites:

Digital Electronics and Logic Design, Computer Organization

List of Experiments

1. 8085 Programming covering all its instructions on microprocessor trainer kit.
2. Interfacing and programming of 8255. (E.g. traffic light controller).
3. Interfacing and programming of 8254.
4. Interfacing and programming of 8279.
5. A/D and D/A converter interface.
6. Stepper motor interface.
7. Display interface.

Suggested Readings:

1. Ramesh S Gaonkar, “Microprocessor Architecture, Programming and Applications with the 8085”, 5th edition, Prentice Hall, 2002.

With effect from Academic Year 2014-15

IT 228

MINI PROJECT - II

Instruction	3 periods per week
Sessional	25 Marks
Credits	1

Course Objectives:

1. To learn by doing, by taking responsibility of the end product.
2. To develop capability to analyse and solve real world problems with an emphasis on applying/integrating knowledge acquired.

Course Outcomes:

Students should be able to do the following:

1. Construct innovative solutions.
2. To work in team as well as individuals.
3. To manage time and resources.

The Students are required to implement one of the projects from project exercise given in the suggested readings of the theory subjects. Focus may be on File structures,

Micro Processor Based Projects, Development of any Controller Circuits using CPLD's or FPGA's. During the implementation of the project, Personnel Software Process (PSP) has to be followed. Report of the project work has to be submitted for evaluation.