
# PUNE VIDYARTHI GRIHA'S

# COLLEGE OF ENGINEERING AND TECHNOLOGY, PUNE-9

**(AFFILIATED TO SAVITRIBAI PHULE PUNE UNIVERSIT, PUNE)**

**DEPARTMENT OF INFORMATIONTECHNOLOGY**

## CURRICULUM BOOK

**ACADEMIC YEAR: 2019-20**

**FOR THE PROGRAMME**

## S.E.INFORMATION TECHNOLOGY


**PUNE VIDYARTHI GRIHA'S**

**COLLEGE OF ENGINEERING AND TECHNOLOGY**

# VISION

**TO ACHIEVE EXCELLENCE IN ENGINEERING EDUCATION**

# MISSION

- **To satisfy all stakeholders**
- **To develop ethical, highly motivated engineering professionals with good human values, requisite skills and competencies**
- **To adopt innovative teaching mechanisms**
- **To promote research culture**
- **To contribute to country's economic development**
- **To be responsive to changes in technology, socio-economic and environmental conditions**

# DEPARTMENT OF INFORMATION TECHNOLOGY

## VISION

**To Empower Students to Face the Technological Challenges of 21st Century by Imparting Quality Education in the Field of Information Technology**

## MISSION

1) **To impart knowledge through innovative teaching-learning process to cater the needs of industries and higher education.**
2) **To inculcate good human values, professional competencies and create awareness about global technologies in the field of Computer Engineering.**
3) **To respond to rapid changes in the field of Information Technology.**

## PROGRAM EDUCATIONAL OBJECTIVES

**PEO1:** Possess strong fundamental concepts in Engineering Science andTechnology to address future technological challenges of Information Technology.

**PEO2:** Possess knowledge and skills in the field of Information Technology forengineering problems with innovative approaches.

**PEO3:** Possess behavioral aspects for research, entrepreneurship and higherstudies in the field of Computer Science and Information Technology.

**PEO4:**Havecommitment to ethical practices in the field of InformationTechnology and, societal contributions through communities and life-long learning.

**PEO5**:Possess better interpersonal and presentation skills to cope up with therapid changes in the field of Information Technology at global level.

## PROGRAMME OUTCOMES

**The Program Outcomes of the Department of Information Technology are:**

**PO1)**An ability to apply knowledge of computing, engineering mathematics, statistics, science, and engineering and technology.

**PO2)**An ability to identify and analyze the problem, provide a systematic solution by conducting experiments, interpreting the data and drawing substantial conclusion.

**PO3)**An ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints.

**PO4)**An ability to identify, formulate, and provide systematic solutions to complex engineering problems and validate the solution.

**PO5)**An ability to apply appropriate resources, skills, modern engineering tools and technologies necessary for practice as a IT professional.

**PO6)**An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems with necessary constraints and assumptions.

**PO7)**An ability to analyze the local and global impact of computing on individuals, organizations and society.

**PO8)**An ability to understand professional, ethical, legal, security and social issues and responsibilities.

**PO9)**An ability to function effectively as an individual or as a team member to accomplish a desired goal(s) in multidisciplinary environment.

**PO10)**An ability to engage in life-long learning and continuing professional development to cope up with fast changes in the technologies/tools with the help of open electives, professional organizations and extra-curricular activities.

**PO11)**An ability to communicate effectively in engineering community at large by means of effective presentations, report writing, paper publications, demonstrations.

**PO12)**An ability to understand engineering, management, financial aspects, performance, optimizations and time complexity necessary for professional practice.

**PO13)**An ability to apply design and development principles in the construction of software systems of varying complexity.

## PROGRAMME SPECIFIC OUTCOMES

At the Completion Graduates will be competently -

**PSO1.** Analyze and develop effective and efficient software solution in the field of data base management system ,web technology ,networking etc. by applying the core concepts of Information Technology.

**PSO2.** Work in teams in various roles to manage IT projects with the help of project management techniques.

# INDEX

# *Second Year*

# *Curriculum Book*

# Syllabus Structure of Savitribai Phule Pune University, Pune

## Semester I

| Subject Code | Subject | Teaching Scheme | | | Examination Scheme | | | | | Total Marks | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Lecture | Tutorial | Practical | Theory Paper | Theory Online | TW | PR | OR | | |
| 214441 | Discrete Structures | 4 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 4 |
| 214442 | Computer Organization & Architecture | 4 | -- | -- | 50 | 50 | | | | 100 | 4 |
| 214443 | Digital Electronics and Logic Design | 4 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 4 |
| 214444 | Fundamentals of Data Structures | 4 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 4 |
| 214445 | Problem Solving and Object Oriented programming | 4 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 4 |
| 214446 | Digital Laboratory | -- | -- | 2 | -- | -- | 25 | 50 | -- | 75 | 1 |
| 214447 | Programming Laboratory | -- | -- | 4 | -- | -- | 25 | 50 | -- | 75 | 2 |
| 214448 | Object Oriented programming Lab. | -- | -- | 2 | -- | -- | 25 | 50 | | 75 | 1 |
| 214449 | Communication Skills | -- | -- | 2 | -- | | 25 | -- | -- | 25 | 1 |
| | Audit Course | -- | -- | -- | -- | -- | -- | -- | -- | Grade | |
| | **Total** | **20** | **--** | **10** | **250** | **250** | **100** | **150** | **--** | **750** | **25** |
| | **Total of Part-I** | **30 Hours** | | | | **750** | | | | | |

## Semester II

| Subject Code | Subject | Teaching Scheme | | | Examination Scheme | | | | | Total Marks | Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Lecture | Tutorial | Practical | Theory Paper | Theory Online | TW | PR | OR | | |
| 207003 | Engineering Mathematics-III | 4 | 1 | -- | 50 | 50 | 25 | -- | -- | 125 | 5 |
| 214450 | Computer Graphics | 3 | - | -- | 50 | 50 | -- | -- | -- | 100 | 3 |
| 214451 | Processor Architecture and Interfacing | 4 | - | - | 50 | 50 | -- | -- | -- | 100 | 4 |
| 214452 | Data Structures &Files | 4 | - | - | 50 | 50 | -- | -- | -- | 100 | 4 |
| 214453 | Foundations of Communication and Computer Network | 4 | - | - | 50 | 50 | -- | -- | -- | 100 | 4 |
| 214454 | Processor Interfacing Laboratory | -- | -- | 4 | -- | -- | 25 | 50 | -- | 75 | 2 |
| 214455 | Data Structure and Files Laboratory | -- | -- | 4 | -- | -- | 25 | 50 | -- | 75 | 2 |
| 214456 | Computer Graphics Laboratory | -- | -- | 2 | -- | -- | 25 | 50 | -- | 75 | 1 |
| | Audit Course | -- | -- | -- | -- | -- | -- | -- | -- | Grade | |
| | **Total** | **19** | **01** | **10** | **250** | **250** | **100** | **150** | **--** | **750** | **25** |
| | **Totalof Part-II** | **30 Hours** | | | | **750** | | | | | |

# *SE IT(2015 Course)*

# *Sem. I*

# DISCRETE STRUCTURES

| | | | |
|---|---|---|---|
| **Course Title:** | **DISCRETE STRUCTURES** | **Course Number:214441** | **Credits: 4** |
| **Designation of Course** | Professional Core | | |
| **Teaching Scheme:** 4 Hrs/Week | | **Laboratories:** -- | |
| **Course Assessment Methods** | **Direct methods** | On-line Examination: 50 Marks | Theory Examination: 50 Marks |
| | | Term-work | Practical/Oral |
| | **Indirect Methods** | Assignments, Presentations | Seminars, Quiz, Q&A session, Group Discussion |
| **Prerequisites** | **1.** Basic Mathematics. | | |
| **Introduction of Course** | | | |
| **Course Objectives** | | | |
| 1 | Use appropriate set, function, or relation models to analyse, practical examples, interpret the associated operations and terminology in context, Apply formal methods of symbolic propositional and predicate logic. | | |
| 2 | Understand the basics of discrete probability and number theory, and be able to apply the methods from these subjects in problem solving. | | |
| 3 | Understand some basic properties of graphs and related discrete structures, and be able to solve these to practical examples. | | |
| 4 | Illustrate by example the basic terminology of graph theory, and some of the properties and special cases of each, demonstrate different traversal methods for trees and graphs, Model problems in computer science using graphs and trees. | | |
| **Course Outcomes** | | | |
| CO1 | Apply formal proof techniques, and explain their reasoning clearly, Construct inductive hypothesis and carry out simple induction proofs | | |
| CO2 | Use graph theoretic models and data structures to model and solve some basic problems in Informatics | | |
| CO3 | Illustrate by example, basic terminology and model problems in computer engineering using graphs and trees and Use graph algorithms for suitable applications. | | |
| CO4 | Demonstrate the knowledge and technical skills to be successful in a specialized, computer-based, graphics field . | | |
| CO5 | Be a master of the solution of linear recurrence equations with constant coefficients and the knowledge of the so called Master Theorem.and understand a master of the concepts injective, surjective and bijective functions | | |
| **Course Contents** | | | |
| **Unit-I** | **SETS AND PROPOSITIONS** | | |
| | Finite and Infinite sets, Un-count ably infinite sets, Principle of inclusion and exclusion, multisets. Propositions, Conditional Propositions, Logical Connectivity Propositional calculus, Universal and Existential Quantifiers, Normal forms methods of proofs, Mathematical Induction. | | |
| **Unit-II** | **RELATIONS AND FUNCTIONS** | | |

| | | | |
|---|---|---|---|
| | Properties of Binary Relations, Closure of relations, Warshall's algorithm, Equivalence Relations and Partitions, Partial ordering relations and lattices , Chains and Anti chains, Functions, Composition of functions, Invertible functions, Generating functions, Job scheduling Problem,<br>Recurrence Relation, Linear Recurrence Relations With constant Coefficients, Homogeneous Solutions, Total solutions, solutions by the method of generating functions. | | |
| **Unit-III** | **GROUPS AND RINGS** | | |
| | Algebraic Systems, Groups, Semi Groups, Monoid, Subgroups, Permutation Groups, Codes and Group Codes, Normal Subgroups, Ring Integral Domain, Field, Ring Homomorphism, Polynomial Rings, Cyclic Codes | | |
| **Unit-IV** | **GRAPH THEORY** | | |
| | Basic terminology, multi graphs and weighted graphs, paths and circuits, Hamiltonian paths and circuits, Euler paths and circuits, factors of a graph, planer graph, shortest path in weighted graph, Travelling salesman problem | | |
| **Unit- V** | **TREES** | | |
| | Trees, rooted trees, path length in rooted trees, prefix codes ,binary search trees, spanning trees and cut set, minimal spanning trees, Kruskal's and Prim's algorithms for minimal spanning tree , The Max flow –Min cut theorem. | | |
| **Unit-VI** | **PERMUTATIONS ,COMBINATIONS AND DISCRETE PROBABILITY** | | |
| | Permutations and Combinations: rule of sum and product, Combinations, Algorithms for generation of Permutations and Combinations, Discrete Probability, Conditional Probability, Bayes'Theorem Information and Mutual Information | | |
| **Text Books** | **Author** | **Title of Book** | **Publication** |
| T1 | C. L. Liu and D. P. Mohapatra, , | "Elements of Discrete Mathematics" | Tata McGraw Hill. |
| T2 | R. Johnsonbaugh, | "Discrete Mathematics" | Pearson Education |
| **Reference Books** | | | |
| R1 | N. Biggs, 3rd Edition, , | "Discrete Mathematics" | Oxford University Press |
| R2 | Kenneth H. Rosen, , | "Discrete Mathematics and its Applications", | McGraw-Hill |
| R3 | B. Kolman, R. Busby and S. Ross, | "Discrete Mathematical Structures" | Pearson Education |
| **Self-Learning Facilities** | NPTELLecture Series by Dr. Kamala Srivastva | | |

## COMPUTER ORGANIZATION AND ARCHITECTURE

| | | | |
|---|---|---|---|
| **Course Title:** | **Computer Organization and Architecture** | **Course Number:** 214442 | **Credits:** 4 |
| **Designation of Course** | Professional Core | | |
| **Teaching Scheme:**4 Hours/Week | | **Laboratories:** NIL | |
| **Course Assessment Methods** | **Direct methods** | Online Examination: 50 Marks | Theory/ Examination:50 Marks |
| | | | |
| | **Indirect Methods** | Assignments, Class Test | Quiz, Q&A session, |
| **Prerequisites** | Fundamental of Programming Languages | | |
| | | | |
| **Introduction of Course: This subject contains structure & function of computer system, parallel organization of multi-processor and multi-core system.** | | | |
| | | | |
| **Course Objectives** | | | |
| 1 | To understand the structure, function & characteristics of computer systems. | | |
| 2 | To understand the design of the various functional units of digital computers. | | |
| 3 | To understand instruction level parallelism & parallel organization of multi-processor & multi core systems | | |
| **Course Outcomes** | | | |
| CO1 | Student will able to Solve problems based on computer arithmetic. | | |
| CO2 | Student will able toExplain processor structure & its functions. | | |
| CO3 | Student will able toObtain knowledge about micro-programming of a processor. | | |
| CO4 | Understand concepts related to memory & IO organization. | | |
| CO5 | Acquire knowledge about instruction level parallelism & parallel organization of multi-processors & multi core systems | | |
| **Course Contents** | | | |
| **Unit-I** | **COMPUTER EVOLUTION, PERFORMANCE MEASUREMENT &ARITHMETIC** | | |
| | A Brief History of Computers, Von Neumann Architecture, Harvard Architecture.<br>Computer Performance Measurement – Benchmarks (SPEC) for Evaluation, Metrics such as CPU Time, Throughput, etc., Aspects & Factors affecting Computer Performance, Comparing Computer Performances, Marketing Metrics – MIPS & MFLOPS, Speedup & Amdahl's Law<br>Booths Algorithm For Signed Multiplication & it's Hardware Implementation, Restoring And Non | | |

| | |
|---|---|
| | Restoring Division Algorithms & it's Hardware Implementation |
| **Unit-II** | **THE CENTRAL PROCESSING UNIT** |
| | Arithmetic & Logic Unit.<br>Instruction Sets: - Machine Instruction Characteristics, Types of Operands and Types of Operations, Addressing Modes, Instruction Formats, Instruction Types<br>Processor Structure and Function - Processor Organization, Register Organization, The Instruction Cycle and Instruction Pipelining.<br>RISC: Instruction Execution Characteristics, RISC Vs CISC, RISC Architecture - MIPS. |
| **Unit-III** | **THE CONTROL UNIT** |
| | Instruction Cycle & Micro Operations, Functional Requirements & Operations of the Control Unit, Block Schematic & Control Signals, Single Bus Processor Organization, Control Signal example with Micro Operations and Register Transfer.<br>Control Unit Design Methods - Hardwired Control – State Table Method, Design example - Multiplier CU.<br>Micro-Programmed Control - Basic Concepts, Microinstructions & Formats, Control Memory, Micro-Programmed Control Unit Schematic, Microinstruction Sequencing - Design Considerations, Sequencing Techniques, Address Generation, Microinstruction Execution - A Taxonomy of Microinstructions, Microinstruction Encoding |
| **Unit-IV** | **MEMORY & I/O ORGANIZATION** |
| | Characteristics of Memory Systems, Internal and External Memory Types.<br>Memory Hierarchy, Principle Of Locality, Cache Memory – Basics, Performance Metrics & Improvements, Organization and Mapping Techniques, Handling Cache Misses & Writes, Replacement Algorithms, Multilevel Caches, Cache Coherence, Snooping & MESI Protocols, Memory Segmentation & Interleaved Memory System.<br>Virtual Memory: Main Memory Allocation, Virtual to Physical Address Translation, Paging, Page Placement & Location, Page Faults, TLB in Address Translation, Handling TLB Misses & Page Faults.<br>Input / Output Systems, Programmed I/O, Interrupt Driven I/O, Direct Memory Access (DMA). |
| **Unit- V** | **INSTRUCTION LEVEL PARALLELISM** |
| | MIPS Implementation Overview, Digital Logic for MIPS Implementation, Single Data path for MIPS Architecture, Simple MIPS Implementation with Control Signals.<br>Overview of Instruction Pipelining, Performance Improvement, MIPS Instruction Set for Pipelining, Pipeline Hazards: Structural, Data – Forwarding & Code Reordering, Control – Branch Prediction, 5 Stage Pipeline with Data path & Control for MIPS Architecture, Graphical Representation of Pipelines, Data Hazards – Forwarding & Stalling for MIPS Pipeline, Control Hazards – Dynamic Branch Prediction & Delayed Branch for MIPS Pipeline. Superscalar Processors. |
| **Unit-VI** | **PARALLEL ORGANIZATION** |
| | Parallel Organization – Multiprocessors, Multicores & Clusters. Flynn's Taxonomy for Multiple Processor Organizations, Closely and Loosely Coupled Multiprocessors Systems, Symmetric Multiprocessor (SMP) Organization, Multithreading – Fine Grained, Coarse Grained & Simultaneous (SMT) Threading, Chip Multiprocessing, Cluster Configuration, UMA, NUMA & CC-NUMA.<br>Multicore Architectures – Hardware & Software Issues in Multicore Organization, Multicore |

| | | | |
|---|---|---|---|
| | Organizations, Intel X86 Multicore Organizations – Core Duo & Core i7. | | |
| **Text Books** | **Author** | **Title of Book** | **Publication** |
| T1 | W. Stallings | Computer Organization and Architecture: Designing for Performance | 8th Edition, Prentice Hall of India, 2010, ISBN 13: 978-0-13-607373-4 |
| T2 | D. Patterson, J. Hennessy | Computer Organization and Design: The Hardware Software Interface | 4th Edition, Morgan Kaufmann, Oct 2013, ISBN 978-0-12-374750-1 |
| **Reference Books** | | | |
| R1 | C.Hamacher, V. Zvonko, S. Zaky | Computer Organization | 5th edition, McGraw Hill, 2002, ISBN: 007-120411-3 |
| R2 | M. Usha, T. S. Srikanth, | Computer System Architecture and Organization | Wiley, 2014, ISBN: 978-81-265-2284-2 |
| R3 | A. S. Tanenbaum | Structured Computer Organization | 4th Edition, Prentice Hall of India, 1991 ISBN: 81–203–1553–7. |
| R4 | G. George | Computer Organization: Hardware and Software | 2nd Edition, Prentice Hall of India, 1986 |
| R5 | J. Hays | Computer Architecture and Organization | 2nd Edition, McGraw-Hill, 1988 ISBN 0–07–100479–3 |
| **Self-Learning Facilities** | Books, PPTs | | |
| | | | |
| **Web Resources** | http://nptel.ac.in/courses/106103068/7 | | |
| | | | |

# DIGITAL ELECTRONICS & LOGIC DESIGN

| | | | |
|---|---|---|---|
| **Course Title:** | **Digital Electronics & Logic Design** | **Course Number:** 214443 | **Credits:** 4 |
| **Designation of Course** | Professional Core | | |
| **Teaching Scheme:**4 Hours/Week | | **Laboratories:**2Hrs/Week | |
| **Course Assessment Methods** | **Direct methods** | Online Examination: 50 Marks | Theory/ Examination:50 Marks |
| | | Term-work | Practical/ |
| | **Indirect Methods** | Assignments, Class Test | Quiz, Q&A session, |
| **Prerequisites** | Basic Electronics Engineering | | |
| | | | |
| **Introduction of Course: This subject contains the design of CLC, SSI, MSI, PLD and VHDL programming** | | | |
| | | | |
| **Course Objectives** | | | |
| 1 | To learn and understand basic digital design techniques | | |
| 2 | To develop design and implementation skills of combinational and sequentiallogic circuits. | | |
| 3 | To introduce digital logic design software such as VHDL Programming. | | |
| **Course Outcomes** | | | |
| CO1 | Spectacle an awareness and apply knowledge of number systems, codes, Boolean algebra and use necessary A.C, D.C Loading characteristics as well as functioning while designing with logic gates. | | |
| CO2 | Use logic function representation for simplification with K-Maps and analyze as well as design Combinational logic circuits using SSI & MSI chips | | |
| CO3 | Analyze Sequential circuits like Flip-Flops (Truth Table, Excitation table), their conversion & design the applications | | |
| CO4 | Identify the Digital Circuits, Input/Outputs to replace by FPGA | | |
| CO5 | Use VHDL programming technique with different modelling styles for any digital circuits | | |
| **Course Contents** | | | |
| **Unit-I** | **NUMBER SYSTEM AND LOGIC FAMILIES** | | |
| | Introduction to digital electronics & Boolean algebra.<br>**Number Systems** - Binary, Octal, Hexadecimal and their conversions.<br>**Signed Binary number representation and Arithmetic's:** Signed & True Magnitude, 1's complement, 2's complement representation and arithmetic's.<br>**Codes:** BCD, Excess-3, Gray code, Binary Code and their conversion.<br>Switching characteristics of BJT & FET, IC Characteristics.<br>**TTL:** Standard TTL characteristics, Operation of TTL NAND gate, Subfamilies, Configurations-Active pull-up, Wired AND, totem pole, open collector.<br>**CMOS:** Standard CMOS characteristics, operation of CMOS NAND, Subfamilies, CMOS configurations Wired Logic, Open drain outputs.<br>Comparison of TTL & CMOS, Interfacing: TTL to CMOS and CMOS to TTL | | |
| | **Practical** | | |
| | | | |

| | |
|---|---|
| **Unit-II** | **COMBINATIONAL LOGIC DESIGN** |
| | **Logic minimization:** Representation of truth-table, SOP form, POS form, Simplification of logical functions, Minimization of SOP and POS forms, don't care Conditions.<br>**Reduction techniques:** K-Maps up to 4 variables and Quine - McClusky technique.<br>**CLC design using SSI chips** – Code converters, Half- Adder, Full Adder, Half Subtractor, Full Subtractor, n bit Binary adder, Look ahead carry generator.Magnitude comparator using IC 7485.<br>**Introduction to MSI functions & chips -** Multiplexers (IC 74151 and IC 74153), Decoder / Demultiplexer (IC 74138), Encoder (IC 74147), Binary adder (IC 7483).<br>**CLC design using MSI chips –** BCD & Excess 3 adder &subtractor using IC 7483, Implementation of logic functions using IC 74151,74153& 74138. |
| | **Practical** |
| | 1. Design & implementation of different code convertors |
| | 2. Implementation of logic functions using multiplexer IC 74153 & decoder IC 74138. |
| **Unit-III** | **SEQUENTIAL LOGIC** |
| | Introduction to sequential circuits. Difference between combinational circuits and sequential circuits, memory element – latch.<br>**Flip- Flops:** Design, truth table, excitation table of SR, JK, D, T flip flops. Study of flip flops with asynchronous and synchronous Preset& Clear, Master Slave configuration, conversion from one type to another type of flip flop. Study of flip flop ICs - 7473, 7474, 7476<br>**Application of flip-flops** – Bounce elimination switch, Counters- asynchronous, synchronous and modulo counters study of modulus n counter ICs- 7490, 74191 & their applications to implement mod counters. |
| | **Practical** |
| | 1. Design (State diagram, state table & K map) and implementation of 3 bit Up and Down Asynchronous and Synchronous Counter using master slave JK flip-flop IC 7476 |
| | 2. Design and implementation of Module 'n' counter with IC7490 and IC 74191. |
| **Unit-IV** | **SEQUENTIAL LOGIC DESIGN** |
| | **Registers**- Buffer register, shift register types - SISO, SIPO, PISO & PIPO, applications of shift registers - ring counter, twisted ring counter, study of universal shift register IC – 74194,<br>Sequence generators using counters & shift register, Pseudo Random Binary Sequence Generator.<br>Basic design steps-State diagram, State table, State reduction, State assignment, Mealy and Moore machines representation, Implementation, finite state machine implementation, sequence detector using Moore & Mealy model. |
| | **Practical** |
| | 1. Design (State Diagram, State Table, K Map) and implementation of Sequence Generator using Shift Register IC 74194 |
| **Unit- V** | **PROGRAMMABLE LOGIC DEVICES AND INTRIDUCTION TO HDL** |
| | **Algorithmic State Machines**- ASM notations, charts (eg- counters, washing machine, lift controller, vending machine), design using multiplexer controller |

| | | | |
|---|---|---|---|
| | method (eg- counters).<br>**Introduction to PLD's –** ROM, PAL, PLA, Design of 4 variable SOP using PLDs, Basic architecture of SPLD and CPLD, Study of CPLD architecture XC9572, Basic architecture of FPGA, CPLD. Design flow (Basic Concept of Simulation and Synthesis)<br>**Introduction to HDL –** Necessity, Characteristics & Types | | |
| | **Practical** | | |
| **Unit-VI** | **VHDL PROGRAMMING** | | |
| | **Introduction to VHDL -** Library, Package, Entity, Architecture, Data Objects (Variable, signal & constant), Data Types (scalar, composite array type & predefined data types, Attributes (necessity and use. 'event attribute). **VHDL Modeling styles –** Dataflow, behavioural & structural<br>**VHDL statements -** Concurrent Statements (With. Select, When..Else), Sequential Statements (if..else, case)<br>**VHDL design Examples -** Multiplexer, binary adder, counter, shift register. | | |
| | **Practical** | | |
| | 1. 4:1 MUX using data flow & structural modeling | | |
| | 2. Full Adder using behavioural & structural modeling | | |
| | 3. 3 bit controlled up/ down synchronous counter with preset& clear. | | |
| **Text Books** | **Author** | **Title of Book** | **Publication** |
| T1 | R.P. Jain | Modern Digital Electronics | Tata McGraw-Hill, ISBN: 0–07–049492–4 |
| T2 | Stephen Brown, ZvonkoVranesic | Fundamentals of Digital Logic with VHDL Design | McGraw-Hill, ISBN: 978–0–07–352953–0 |
| | | | |
| **Reference Books** | | | |
| R1 | Flyod, | "Digital Principles | Pearson Education ISBN:978-81- 7758-643-6. |
| R2 | M Morris Mano | Digital Design | Prentice Hall, 3rd Edition, ISBN: 0130621218. |
| R3 | John Yarbrough, | Digital Logic applications and Design | Thomson Publication<br>ISBN: 978-0314066756 |
| R4 | Malvino, D. Leach, | Digital Principles and Applications | 5th edition, Tata McGraw Hill |
| R5 | J.Bhaskar | VHDL Primer | , Pearson Education,3rd Edition, ISBN: 0071226249 |
| R6 | Kohavi Z., Jha N.K,, | Switching and Finite Automata Theory | Cambridge University Press, India, 2nd Edition, ISBN: 978-0-521-85748-2 |
| **Self-Learning Facilities** | Books, PPTs | | |
| | | | |
| **Web Resources** | http://nptel.ac.in/courses/117106086/1 | | |

| | |
|---|---|
| **Assignments** | |
| 1 | Design & implementation of different code convertors |
| 2 | Implementation of logic functions using multiplexer IC 74153 & decoder IC 74138. |
| 3 | Design (State diagram, state table & K map) and implementation of 3 bit Up and Down Asynchronous and Synchronous Counter using master slave JK flip-flop IC 7476 |
| 4 | Design and implementation of Module 'n' counter with IC7490 and IC 74191. |
| 5 | Design (State Diagram, State Table, K Map) and implementation of Sequence Generator using Shift Register IC 74194 |

# FUNDAMENTALS OF DATA STRUCTURES

| | | | |
|---|---|---|---|
| **Course Title:** | **Fundamental Of Data Structures** | **Course Number:214444** | **Credits: 4** |
| **Designation of Course** | Professional Core | | |
| **Teaching Scheme:** 4 Hrs/Week | | **Laboratories:** 4 Hrs/Week | |
| **Course Assessment Methods** | **Direct methods** | On-line Examination: 50 Marks | Theory/End Semester Examination: 50 Marks |
| | | Term-work 25 Marks | Practical/Oral 50 Marks |
| | **Indirect Methods** | | |
| **Prerequisites** | Fundamental knowledge of 'C' and basics of algorithms | | |
| **Introduction of Course** | | | |
| **Course Objectives** | | | |
| 1 | To learn C language constructs and pointers in depth. | | |
| 2 | To learn algorithm development and analysis of algorithms. | | |
| 3 | To learn linear data structures and their applications | | |
| 4 | To learn different searching and sorting techniques | | |
| **Course Outcomes** | | | |
| CO1 | Student will be able to apply appropriate constructs of C language, coding standards for application development. | | |
| CO2 | Students will be to use dynamic memory allocation concepts and file handling in various application developments. | | |
| CO3 | Students will be able to perform basic analysis of algorithms with respect to time and space complexity. | | |
| CO4 | Students will be able to select appropriate searching and/or sorting techniques in the application development. | | |
| CO5 | Students will be able to select and use appropriate data structures for problem solving and Programming. | | |
| CO6 | Students will be able to use algorithmic foundations for solving problems and programming. | | |
| **Course Contents** | | | |
| **Unit-I** | **C BASICS** | | |
| | Control structures, arrays, functions and parameter passing Structure and Union, String manipulation, matrix operations | | |
| | **Practical** | | |
| | 1. Represent sets using one dimensional arrays and implement functions to perform<br>i. Union<br>ii. Intersection<br>iii. Difference<br>i. iv. Symmetric difference of two sets | | |
| | 2. Create a Database using array of structures and perform following operations on it:<br>i. Create Database<br>ii. Display Database<br>iii. Add record<br>iv. Search record | | |

| | |
|---|---|
| | v. Modify record<br>vi. Delete record |
| **Unit-II** | **POINTERS IN C AND FILE HANDLING** |
| | Introduction to Pointers, dynamic memory allocation, pointer to pointer, pointer to single and multidimensional arrays, array of pointers, string and structure manipulation using pointers, pointer to functions. Pointer to file structure and basic operations on file, file handling in C. |
| | **Practical** |
| | 1. Represent matrix using two dimensional arrays and perform following operations **with and without pointers**:<br>i. Addition<br>ii. multiplication<br>iii. transpose<br>iv. Saddle Point |
| | **2.** Implement following operations on string **with / without pointers (without using library functions)**<br>i. Length<br>ii. Palindrome<br>iii. String comparison<br>iv. Copy<br>v. Reverse<br>vi. Substring |
| | 3. Implement sequential file and perform following operations:<br>i. Display<br>ii. Add records<br>iii. Search record<br>iv. Modify record<br>v. Delete record |
| **Unit-III** | **INTRODUCTION TO DATA STRUCTURES AND ANALYSIS OF ALGORITHMS** |
| | Introduction to Data Structures: Concept of data, Data object, Data structure, Abstract Data Types, realization of ADT in 'C'. Concept of Primitive and non-primitive, linear and Non-linear, static and dynamic, persistent and ephemeral data structures.<br>Analysis of algorithm: frequency count and its importance in analysis of an algorithm, Time complexity & Space complexity of an algorithm, Big 'O', 'Ω' and 'Θ' notations, Best, Worst and Average case analysis of an algorithm. |
| | **Practical** |
| **Unit-IV** | **SEARCHING AND SORTING TEHNIQUES** |
| | Need of searching and sorting, Concept of internal and external sorting, sort stability.<br>Searching methods: Linear and binary search algorithms their comparison and complexity analysis<br>Sorting methods: Bubble, selection, insertion, merge, quick, bucket sort and their time and space complexity analysis |
| | **Practical** |
| | 1. Sort the set of strings in ascending order using Bubble sort and descending order by using Selection sort or Insertion sort. (Display pass by pass output)<br>b) Search a particular string using binary search **with and without recursion.** |

| | |
|---|---|
| | **2.** Implement Quick Sort / **Merge Sort** to sort the given list of numbers. Display corresponding list in each pass. (with and without recursion) |
| **Unit- V** | **LINEAR DATA STRUCTURES USING SEQUENTIAL ORGANIZATION** |
| | Concept of sequential organization, Concept of Linear data structures, Concept of ordered list, Multidimensional arrays and their storage representation: row major and column major form and address calculation. Representation of sparse matrix using arrays, algorithms for sparse matrix addition, simple and fast transpose, polynomial representation using arrays. Analysis of these algorithms. Introduction to Stack and Queue, and their implementation using sequential organization, use of stack in recursion. |
| | **Practical** |
| | 1. Accept conventional matrix and convert it into sparse matrix using structure and perform **addition**, simple and fast transpose |
| **Unit-VI** | **LINEAR DATA STRUCTURES USING LINKED ORGANIZATION** |
| | Concept of linked organization, singly linked list, doubly linked list, circular linked list. Linked list as an ADT. Representation and manipulations of polynomials using linked lists, comparison of a sequential and linked memory organization, concept of Generalized Linked List, representation polynomial using GLL. |
| | **Practical** |
| | 1. Implement a singly linked list with following options<br>i. Insertion of a node at any location<br>ii. Deletion of a node from any location<br>iii. display a list<br>iv. Display in reverse<br>v. Revert the list without using additional data structure. |
| | 2. Implement polynomial using CLL and perform<br>i. Addition of Polynomials<br>ii. Multiplication of polynomials and<br>iii. Evaluation of polynomial |
| | 3. Implement any database using doubly linked list with following options<br>i. Insert a record<br>ii. delete a record<br>iii. modify a record<br>iv. Display list forward<br>v. Display list backward |
| | 4. Implement Generalized Linked List to create and display the book index. |

# PROBLEM SOLVING AND OBJECT ORIENTED PROGRAMMING

| | | | |
|---|---|---|---|
| **Course Title:** | **Problem Solving and Object Oriented Programming** | **Course Number:** 214445 | **Credits:** 4 |
| **Year:** SE | | **Semester:** I | |
| **Designation of Course** | | Professional Core | |
| **Teaching Scheme:** 4 Hrs/Week | | **Tutorial:** | |
| **Course Assessment Methods** | **Direct methods** | On-line Examination: 50 Marks | End Semester Examination: 50 Marks |
| | | | Practical/Oral/Term Work |
| | **Indirect Methods** | Assignments, Presentations | Seminars, Quiz, Q&A session, Group Discussion |
| **Prerequisites** | Principles of Programming Languages, Fundamentals of Data Structures | | |
| **Course Objectives** | | | |
| 1 | Employ a problem-solving strategy to breakdown a complex problem into a series of simpler tasks. | | |
| 2 | Execute problem-solving actions appropriate to completing a variety of sub problems. | | |
| 3 | .Apply analytical and logical thinking to extract facts from a problem description and determine how they relate to one another and to the problems to be solved. | | |
| 4 | Design and implement an object oriented solution to solve a real life problem | | |
| 5 | Develop problem-solving and programming skills using OOP concept. | | |
| **Course Outcomes** | | | |
| CO1 | After studying this subject student should be able to<br>Break a problem into logical pieces and develop algorithms for solving simple problems. | | |
| CO2 | Abstract data and entities from the problem domain, build object models and design software solutions using object-oriented principles and strategies. | | |
| CO3 | Discover, explore and apply tools and best practices in object-oriented programming. | | |
| CO4 | Develop programs that appropriately utilize key object-oriented concepts. | | |
| **Course Contents** | | | |
| **Unit-I** | **Problem Solving Concepts ( 6 Hours)** | | |
| | General Problem Solving Concepts-Types of problems, problems solving with computers, difficulties with problem solving, Problem Solving Aspects, Problem Solving Concepts for computer- constants and variables, data types, functions, operators, expressions and equations, Programming Concepts – communicating with computers, organizing the problem, using the tools, testing the solution, coding the program, Top down design | | |
| **Unit-II** | **Problem Solving with Logic Structures (6 Hours)** | | |
| | Programming Structure - modules and their functions, cohesion & Coupling, Local and global variable, parameters, return values, variable names and data dictionaries, four logic structures. Problem solving with sequential logic structure - The sequential logic structure, solution development. Problem Solving with Decisions – decision logic structure, multiple if/then/else instructions, straight-through logic, positive logic, negative logic, logic conversion, decision tables. Problem solving with loops | | |

| | |
|---|---|
| | and case logic structures |
| **Unit-III** | **Foundations of Object Oriented Programming (6 Hours)** |
| | **Introduction**: Introduction to procedural, modular, object-oriented and generic programming<br>techniques, Limitations of procedural programming, Need of object-oriented programming,<br>fundamentals of object-oriented programming: objects, classes, data members, methods, messages, data encapsulation, data abstraction and information hiding, inheritance, polymorphism **++ Extensions to C** : Variable declarations, global scope, 'const', reference variables, operators in C++(scope resolution, new , delete), dynamic memory allocation, function prototypes, default and constant arguments, 'cin', 'cout', inline functions<br>**Class**: Defining a class, data members and member functions, public, private and protected members, inline member functions, static data members, static member functions, constructors, destructors, array of objects, classes, objects and memory, class as ADTs and code reuse |
| **Unit-IV** | **Overloading and Inheritance (8 Hours)** |
| | Function overloading, friend function, friend class<br>**Operator Overloading** : Introduction, Need of operator overloading, rules for operator overloading, overloading the unary and binary operators using member function, operator overloading using friend function, overloading relational and logical operators, overloading new, delete and assignment operator, type conversions<br>**Inheritance** : Introduction, Need of inheritance, base and derived classes, member access control, types of inheritance, derived class constructor, constructors in multiple inheritance, overriding member functions, ambiguity in multiple inheritance, virtual base class |
| **Unit- V** | **Virtual Functions and Templates (7 Hours)** |
| | **Virtual functions** : Pointers to objects, 'this' pointer, Pointers to derived class, virtual function, rules for virtual function, pure virtual function, abstract class, virtual destructors, early and late binding, container classes<br>**Templates** : Introduction, Function template and class template, overloading function template,<br>member function templates and template arguments, Introduction to Standard Template Library (STL), containers, iterators and algorithms |
| **Unit-VI** | **Exception Handling and File I/O (7 Hours)** |
| | **Namespaces**: Introduction, Rules of namespaces<br>**Exception Handling**: Introduction, Exception handling mechanism: try, catch and throw, Multiple Exceptions, Exceptions with arguments<br>**Managing Console I/O Operations**: Introduction, C++ streams, stream classes, unformatted I/O, formatted I/O and I/O manipulators<br>**File I/O**: Introduction, Classes for file stream operations, file operations (open, close, read, write,<br>detect end of file), file modes, File pointers and their manipulations, error handling during file<br>operations |

| **Text Books** | **Author** | **Title of Book** | **Publication** |
|---|---|---|---|
| T1 | R G Dromey | How to Solve it by Computer | Pearson Education |

| | | | |
|---|---|---|---|
| T2 | Maureen Spankle | Problem Solving and Programming Concepts | Pearson, 2011, ISBN-13: 978-0132492645. |
| T3 | . Robert Lafore | Object-Oriented Programming in C++ | SAMS Techmedia. |
| **Reference Books** | | | |
| R1 | Joyce Farrell | Programming Logic and Design | Cengage Learning, |
| R2 | . E. Balaguruswamy | Object-oriented Programming with C++ | Tata McGraw Hill |
| R3 | Herbert Schildt | C++: The Complete Reference | McGraw-Hill. |
| R4 | Kogen | Object Oriented Programming Methodology | Wiley |
| R5 | Venugopal | Mastering C++ | McGraw-Hill |

# *SE IT (2015Course)*

# *Sem. II*

# COMPUTER GRAPHICS

| | | | |
|---|---|---|---|
| **Course Title:** | **COMPUTER GRAPHICS** | **Course Number:214449** | **Credits: 3** |
| **Designation of Course** | Professional Core | | |
| **Teaching Scheme:** 4 Hrs/Week | | **Laboratories:** 2 Hrs/Week | |
| **Course Assessment Methods** | **Direct methods** | On-line/In-sem Examination: 15/50 Marks | Theory/End Semester Examination: 20/50 Marks |
| | | Term-work10/25 | Practical/Oral 20/50 |
| | **Indirect Methods** | Assignments, Presentations | Seminars, Quiz, Q&A session, |
| **Prerequisites** | 1. Computer Programming and basic data structures.<br>2. Mathematics topics such as analytical geometry, trigonometry, linear algebra and matrices.<br>3. Knowledge of vector space, Matrices, Dot products and distances. | | |
| **Introduction of Course** | | | |
| **Course Objectives** | | | |
| 1 | To acquaint the learners with the basic concepts of Computer Graphics. | | |
| 2 | To learn the various algorithms for generating and rendering graphical figures. | | |
| 3 | To get familiar with mathematics behind the graphical transformations. | | |
| 4 | To understand and apply various methods and techniques regarding projections, animation, shading, illumination and lighting. | | |
| 5 | To acquaint the learners with the basic concepts of Computer Graphics. | | |
| 6 | To learn the various algorithms for generating and rendering graphical figures. | | |
| **Course Outcomes** | | | |
| CO1 | Apply mathematics and logic to develop Computer programs for elementary graphicoperations . | | |
| CO2 | Develop scientific and strategic approach to solve complex problems in the domain ofComputer Graphics. | | |
| CO3 | Develop the competency to understand the concepts related to Computer Vision and Virtual reality. | | |
| CO4 | Apply the logic to develop animation and gaming programs. | | |
| **Course Contents** | | | |
| **Unit-I** | **UNIT – I BASIC CONCEPTS**<br>Introduction to Computer Graphics, Basics of graphics systems, Raster scan & Random scan displays,basic display processor<br>**Display Files:** display file structure, algorithms and display file interpreter. Primitive operations on display file<br>**Plotting Primitives:** Scan conversions, lines, line segments, vectors, pixels and frame buffers, vector generation<br>**Line drawing Algorithms:** DDA, Bresenham<br>**Circle drawing Algorithms: -** DDA, Bresenham<br><br>**Character Generation:** Stroke Principle, Starburst Principle, Bit map method, | | |

| | |
|---|---|
| | Introduction to aliasing and anti-aliasing |
| | |
| | **Practical** |
| | To implement DDA Line and Bresenham's Line algorithm. |
| | To implement DDA,Bresenham and Mid point circle algorithm |
| | To Draw various patterns using Line drawing and Circle algorithm. |
| **Unit-II** | **POLYGONS AND GRAPHICAL TRANSFORMATIONS** |
| | **Polygon and its types, inside test, polygon filling methods:** Seed fill, Scan Line, Flood fill and Boundary fill<br><br>**2D Geometric Transformations -** translation, scaling, rotation, other transformations such as reflection, shearing, matrix representation and homogeneous coordinate system, Composite transformations |
| | |
| | **Practical** |
| | To implement seed fill algorithm. |
| | To apply Basic and composite transformation on 2-D object |
| | To draw a 4 X 4 chessboard and rotate it. |
| **Unit-III** | **3D TRANSFORMATIONS AND PROJECTIONS** |
| | Translation, scaling, rotation, rotation about X, Y, Z and arbitrary axis reflection about XY, YZ, XZ and<br>arbitrary plane**.**<br>**Projections:** Types Parallel - Oblique: Cavalier, Cabinet and orthographic :Isometric, Dimetric**,** Trimetricand Perspective - Vanishing Points as 1 point, 2 point and 3 point |
| | |
| | **Practical** |
| | To peform transformation on 3-D object and projections. |
| | |
| **Unit-IV** | **SEGMENTS, WINDOWING AND CLIPPING** |
| | **Segment:** Introduction, Segment table, Segment creation, closing, deleting and renaming, Visibility<br>**Windowing**: Concept of window and viewport, viewing transformations<br>**Line Clipping:** Cohen Sutherland Method, Midpoint subdivision method<br><br>**Polygon Clipping :** Sutherland Hodgman method for clipping convex and concave polygons |
| **Unit- V** | **SHADING, ANIMATION AND GAMING** |
| | **Shading**: Halftoning, Gouraud and Phong Shading<br>**Computer Animation**: Animation sequences, functions & Languages, Key-frame Systems, Motion Specifications.<br>**Gaming platforms**: Graphics Memory Pipeline, Block diagram of NVIDIA workstation and i860<br>Introduction to OpenGL ES |
| | **Practical** |
| | To design animation .- |
| **Unit-VI** | **Curves and Fractals** |

| | | | |
|---|---|---|---|
| | Introduction, Curve generation, Interpolation, interpolating algorithms, interpolatingpolygons, B-Splines and corners, Bezier curves, Fractals, fractal lines and surfaces(With complete mathematical treatment of this unit) Interactive Graphics & usage of atleast two tools of computer graphics Maya, Similar tools) | | |
| | | | |
| | **Practical** | | |
| | To generate Koch curve . | | |
| **Text Books** | **Author** | **Title of Book** | **Publication** |
| T1 | Steven Harrington | Computer Graphics A Programming approach | Tata McGraw Hill. |
| T2 | Davis Rogers | Procedural Elements for Computer Graphics | Tata McGraw Hill |
| | | | |
| **Reference Books** | | | |
| R1 | M Paulin Baker | Computer Graphics | Pearson Education |
| R2 | Zhigang Xiang, Roy Plastock | Computer Graphics | Schaum's Series outlines |
| R3 | Er. Rajiv Chopra | Computer Graphics | S Chand & Company Ltd. |
| R4 | Shirley, Marschner | Fundamentals of Computer Graphics | A K Peters SPD |
| R5 | F.S. Hill JR | Computer Graphics Using Open GL | Pearson Education |
| R6 | Shirley | Computer Graphics | Cengage lerning publication |
| R5 | | | |
| **Self-Learning Facilities** | NPTELLecture Series by Dr.Sukhendu Das | | |
| | | | |
| **Web Resources** | www.opengl.org | | |
| | | | |

# PROCESSOR ARCHITECTURE AND INTERFACING

| | | | |
|---|---|---|---|
| **Course Title:** | **Processor Architecture and Interfacing** | **Course Number:214451** | **Credits: 4** |
| **Designation of Course** | Professional Core | | |
| **Teaching Scheme:** 4 Hrs/Week | | **Laboratories:** 4 Hrs/Week | |
| **Course Assessment Methods** | **Direct methods** | Online Examination: 50 Marks | Theory Examination: 50 Marks |
| | | Term-work | Practical |
| | **Indirect Methods** | Presentations | Objective test |
| **Prerequisites** | *Computer Organization* | | |
| **Introduction of Course:** | | | |
| **Course Objectives** | | | |
| 1 | To learn the architecture and assembly language programming of 80386 Microprocessor. | | |
| 2 | To study architecture and programming of 8051 micro-controllers | | |
| **Course Outcomes** | | | |
| CO1 | Understand microprocessor architecture and Use of segment descriptors, privileges | | |
| CO2 | Use of TSS for handling of multiple task and To classify the interrupts handling in protected mode | | |
| CO3 | Use of instruction sets and addressing modes for writing 8051 programming | | |
| CO4 | Use of timers and serial communication ports for 8051 programming | | |
| **Course Contents** | | | |
| **Unit-I** | **INTRODUCTION TO ASSEMBLY LANGUAGE PROGRAMMING AND 80386 PROCESSORS** | | |
| | Introduction to assembly language programming, ALP tools- Assembler, Linker, Loader, Debugger, Emulator, Assembler directives, Far and near procedure, Macros, DOS Internals, DOS Calls.<br>80386 - Features and Architecture, Register Set, 80386 Real mode segmentation and Address translation, Addressing modes, Instruction set. | | |
| **Unit-II** | **80386 MEMORY MANAGEMENT** | | |
| | Pin Description of 80386, 16/32-bit data transfer mechanism, Pipelined & Non pipelined bus cycles.<br>Segmentation - support registers and Data structures, Descriptors, Memory management through segmentation, Logical to linear/physical address translation.<br>Privileged instructions, Protection in segmentation, Inter-privilege level transfer using Call gates and confirming code segment. | | |
| **Unit-III** | **80386 – PRIVILEGE PROTECTION, MULTITASKING & INTERRUPTS, EXCEPTIONS** | | |
| | Paging - support registers and Data structures, Descriptors, Linear to physical address translation, Page level protection.<br>Multitasking - Support registers and Data structures, Descriptors, Task switching.<br>Real and Protected mode Interrupt structure - IVT, IDT, Type of exceptions and Processing. | | |
| **Unit-IV** | **INTRODUCTION TO 8051 MICROCONTROLLER** | | |
| | Difference between microprocessor and microcontroller, 8051 microcontroller - Features, Architecture, Pin Description. | | |

| | | | |
|---|---|---|---|
| | On-Chip data memory and program memory organization - Register set, Register bank and Special Function Registers (SFRs).<br>Addressing modes, Instruction set. External data memory and program memory organization. | | |
| **Unit- V** | **PORTS, INTERRUPTS & TIMERS/COUNTERS OF 8051** | | |
| | I/O ports programming - Structures, Related SFRs and Configuration.<br>Interrupt programming - Structure and Response, Related SFRs and Configuration.<br>Timers/counters programming - Structure, Related SFRs, Operating modes, Delay calculations and Configuration.<br>Serial port programming - Related SFRs, Operating modes, Baud rate calculation and Configuration. | | |
| **Unit-VI** | **8051 INTERFACING & APPLICATIONS** | | |
| | PPI 8255 – Features, Architecture, Operating modes & Programming.<br>Interfacing of displays: LED, LCD, Seven segments.<br>Keyboard Interfacing, Interfacing of ADC and DAC, Interfacing of stepper motor, Interfacing of Sensors (temperature, pressure), External data memory and program memory interfacing, Design of minimum system using 8051 micro-controller for various applications. | | |
| **Text Books** | **Author** | **Title of Book** | **Publication** |
| T1 | James Turley | Advanced 80386 Programming Techniques | McGraw Hill Education |
| T2 | Kenneth Ayala | The 8051 Micro Controller | Delmar Cengage Learning |
| **Reference Books** | | | |
| R1 | I. Scott MacKenzie , Raphael Chung-Wei Phan | "8051 Microcontroller | Prentice Hall |
| R2 | Tribel Singh | 8088 /8086 Processor | PHI |
| R3 | MazidiM.Gillipse J. | The 8051 Microcontroller and Embedded Systems | Pearson education |
| **Self-Learning Facilities** | Reference and Text Books. | | |
| **Research papers for reference** | **Author** | **Title of Paper** | **Journal/Transaction** |
| 1 | SAMEERA A'AMER ABDUL-KADER | EMULATION OF THE MICROPROCESSOR INTEL 80386 | Diyala Journal of Engineering Sciences, Vol. 02, No. 01, June 2009 |
| **Contents beyond Syllabus** | Introduction of 8086 and programming of 8086, Design of minimum system using 8051 micro-controller<br>for various applications. | | |
| **Additional Experiments** | 1. Execution of all instruction sets of 8086.<br>2. Addition of byte and word array.<br>3. Block Transfer | | |
| **Bridging Courses** | | | |

| Assignments | |
|---|---|
| 1 | Write ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for<br>i) HEX to BCD ii) BCD to HEX iii) EXIT.<br>Display proper strings to prompt the user while accepting the input and displaying the result. |
| 2 | Write ALP to perform string manipulation to calculate string length and reverse a string. The strings to be accepted from the user is to be stored in code segment Module_1 and write FAR PROCEDURES in code segment Module_2 for following operations on the string:<br>i) Concatenation of two strings<br>ii) Compare two strings<br>iii) Number of occurrences of a sub-string in the given string<br>iv) Find number of words, characters, number of lines and number of capital letters from the given text in the data segment<br>**Note**: Use PUBLIC and EXTERN directive. Create .OBJ files of both the modules and link them to create an EXE file. |
| 3 | Write following programs in C using int86, int86x, intdos, intdosx functions<br>i. To delete a file<br>ii. To create a directory<br>iii. Read and display disk information such as Drive, tracks, sectors etc |
| 4 | Study of 80386 architecture (functional diagram, register set and addressing modes) |
| 5 | Write ALP to switch from real mode to protected mode and back to real mode. |
| 6 | Classify the protected mode exceptions as per the vector. |
| 7 | Write a program to add n, 8 bits numbers found in internal ram location 40H onwards and store results in R6 and R7. |
| 8 | Write a program to multiply 16 bit number by 8 bit number and store the result in internal memory location |
| 9 | Write a program for the block transfer (external to internal memory) |
| 10 | Timer programming: ISR based<br>Write ALP to generate 2 KHz square wave using Timer interrupt on any port pin. |
| 11 | Write ALP to interface 8051 with Interface stepper motor to 8051 and write a program to rotate motor with different step angles and with different speeds. |
| | |

## DATA STRUCTURES & FILES

| | | | |
|---|---|---|---|
| **Course Title:** | **Data Structures and Files** | **Course Number:214455** | **Credits: 4** |
| **Designation of Course** | Professional Core | | |
| **Teaching Scheme:** 4 Hrs/Week | | **Laboratories:** 4 Hrs/Week | |
| **Course Assessment Methods** | **Direct methods** | On-line Examination: 50 Marks | Theory/End Semester Examination: 50 Marks |
| | | Term-work 25 Marks | Practical/Oral 50 Marks |
| | **Indirect Methods** | Assignments, | Seminars, Quiz, Q&A session, |
| **Prerequisites** | Fundamentals of Data Structures, Discrete Structures | | |

| | |
|---|---|
| **Introduction of Course** | |
| **Course Objectives** | |
| 1 | 1. To study data structures and their implementations using OOP (C++) and their applications. |
| 2 | 2. To study some advanced data structures such as trees, graphs and tables. |
| 3 | 3. To learn different file organizations. |
| **Course Outcomes** | |
| CO1 | After successful completion of this course, student will be able to<br>1. Analyze algorithms and to determine algorithm correctness and time efficiency class. |
| CO2 | 2. Understand different advanced abstract data type (ADT) and data structures and their implementations. |
| CO3 | 3. Understand different algorithm design techniques (brute -force, divide and conquer, greedy, etc.) and their implementation |
| CO4 | 4. Apply and implement learned algorithm design techniques and data structures to solve problems. |
| **Course Contents** | |
| **Unit-I** | **STACKS AND QUEUES** |
| | Concept of stack, stack as ADT, Implementation of stack using linked organization. Concept of implicit and explicit stack, Applications of stack. Concept of queues as ADT, Implementation of queue using linked organization. Concept of circular queue, double ended queue, Priority queue. Applications of queues. |
| | **Practical** |
| | 1. Implement stack as an abstract data type using linked list and use this ADT for conversion of infix expression to postfix, prefix and evaluation of postfix and prefix expression. |

| | |
|---|---|
| | 2. Implement priority queue as ADT using single linked list for servicing patients in an hospital with priorities as i) Serious (top priority) ii) medium illness (medium priority) iii) General (Least priority). |
| **Unit-II** | **TREES** |
| | Difference in linear and non-linear data structure, Trees and binary trees-concept and terminology. Expression tree. Conversion of general tree to binary tree. Binary tree as an ADT. Recursive and non-recursive algorithms for binary tree traversals, Binary search trees, Binary search tree as ADT, Applications of trees |
| | **Practical** |
| | 1. Create Binary tree and perform following operations:<br>a. Insert<br>b. Display<br>c. Depth of a tree<br>d. Display leaf-nodes<br>e. Create a copy of a tree |
| | 2. Construct and expression tree from postfix/prefix expression and perform recursive and non- recursive In-order, pre-order and post-order traversals. |
| | 3. Implement binary search tree and perform following operations:<br>a. Insert<br>b. Delete<br>c. Search<br>d. Mirror image<br>e. Display<br>f. Display level wise |
| **Unit-III** | **GRAPHS** |
| | Graph as an ADT, Representation of graphs using adjacency matrix and adjacency list, Depth First Search and Breadth First Search traversal. Prim's and Kruskal's algorithms for minimum spanning tree, shortest path using Warshall's and Dijkstra's algorithm, topological sorting. |
| | **Practical** |
| | 1. Consider a friends' network on face book social web site. Model it as a graph to represent each node as a user and a link to represent the fried relationship between them. Store data such as date of birth, number of comments for each user.<br>Find who is having maximum friends<br>Find who has post maximum and minimum comments<br>Find users having birthday in this month.<br>Hint: (Use adjacency list representation and perform DFS and BFS traversals) |
| | **2.** Represent any real world graph using adjacency list /adjacency matrix find minimum spanning tree using Kruskal's algorithm. |
| | **3.** Represent a given graph using adjacency matrix /adjacency list and find the shortest path using Dijkstra's algorithm (single source all destination). |
| | 4. A business house has several offices in different countries; they want to lease phone lines to connect them with each other and the phone company charges different rent to connect different pairs of cities. Business house |

| | |
|---|---|
| | want to connect all its offices with a minimum total cost. Solve the problem by suggesting appropriate data structures . |
| **Unit-IV** | **TABLES** |
| | Symbol Table: Notion of Symbol Table, OBST, Huffman's algorithm, Heap data structure, Min and Max Heap, Heap sort implementation, applications of heap Hash tables and scattered tables: Basic concepts, hash function, characteristics of good hash function, different key-to-address transformations techniques, synonyms or collisions, collision resolution techniques- linear probing, quadratic probing, rehashing, chaining without replacement and chaining with replacement |
| | **Practical** |
| | 1. Store data of students with telephone no and name in the structure using hashing function for telephone number and implement chaining with and without replacement. |
| **Unit- V** | **ADVANCE TREES** |
| | Concept of threaded binary tree. Preorder and In-order traversals of in-order threaded binary tree, Concept of red and black trees, AVL Trees, B trees, B+ trees, Splay trees |
| **Unit-VI** | **FILE ORGANIZATION** |
| | External storage devices, File, File types and file organization (sequential, index sequential and Direct access), Primitive operations and implementations for each type and comparison |
| | **Practical** |
| | 1. Department maintains a student information. The file contains roll number, name, division and address. Write a program to create a sequential file to store and maintain student data. It should allow the user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If student record is found it should display the student details. |
| | 2. 12. Implement direct access file using hashing ( chaining without replacement) perform following operations on it<br>a. Create Database<br>b. Display Database<br>c. Add a record<br>d. Search a record<br>e. Modify a record |

| **Text Books** | **Author** | **Title of Book** | **Publication** |
|---|---|---|---|
| T1 | R. Gilberg, B. Forouzan | Data Structures: A pseudo code approach with C++ | Cengage Learning, ISBN 9788131503140. |
| T2 | E. Horowitz, S. Sahni, D. Mehta, | Fundamentals of Data Structures in C++ | Galgotia Book Source, New Delhi, 1995, ISBN16782928 |
| **Reference Books** | | | |
| R1 | G. A.V, PAI | Data structures and | Mc Graw Hill, ISBN -13: 978-0- |

| | | | |
|---|---|---|---|
| | | Algorithms | 07-066726-6 |
| R2 | Y. Langsam, M. Augenstin, A. Tannenbaum, | Data Structures using C and C++ | Prentice Hall of India, 2002, ISBN-81-203-1177-9. |
| R3 | J. Tremblay, P. Soresan | An introduction to data structures with Applications | Tata McGraw Hill International Editions, 1984, ISBN-0-07-462471- |
| R4 | M. Weiss, | Data Structures and Algorithm Analysis in C++ | Pearson Education, 2002, ISBN-81-7808-670-0 |
| R5 | Folk, Zoellick, Riccardi | File Structures | Pearson Education, ISBN-81-7758-373-5 |
| **Self-Learning Facilities** | NPTEL Lecture Series Dr.Navin Garg | | |
| | | | |
| **Web Resources** | MIT OCW presentations and videos | | |

| **Research papers for reference** | **Author** | **Title of Paper** | **Journal/Transaction** |
|---|---|---|---|
| | William Pugh | Skip Lists: A Probabilistic Alternative to Balanced Trees | Communications of the AGM June 1990 Volume 33 Number 6 |

# FOUNDATIONS OF COMMUNICATION AND COMPUTER NETWORK

| | | | |
|---|---|---|---|
| **Course Title:** | **Foundations of communication and Computer Network** | **Course Number:214453** | **Credits: 4** |
| **Designation of Course** | Professional Core | | |
| **Teaching Scheme:**4Hrs/Week | | **Laboratories:** | |
| **Course Assessment Methods** | **Direct methods** | In-sem Examination: 25+25=50 Marks(online) | Theory/End Semester Examination: 50 Marks |
| | | Term-work | Practical/Oral |
| | **Indirect Methods** | Assignments, Presentations | Seminars, Quiz, Q&A session, Group Discussion |
| **Prerequisites** | NA | | |
| **Introduction of Course** | | | |
| **Course Objectives** | | | |
| 1 | To learn fundamentals of communication system | | |
| 2 | To acquaint themselves with layered model used computer networks | | |
| | | | |
| **Course Outcomes** | | | |
| CO1 | Understand data/signal transmission over communication media | | |
| CO2 | Recognize usage of various modulation techniques in communication | | |
| CO3 | Analyze various spread spectrum and multiplexing techniques | | |
| CO4 | Use concepts of data communication to solve various related problems | | |
| CO5 | Understand error correction and detection techniques. | | |
| CO6 | Acquaint with transmission media and their standards | | |
| | | | |
| **Course Contents** | | | |
| **Unit-I** | **INTRODUCTION TO COMMUNICATION AND NETWORKING** **9 Hours** | | |
| | **Introduction To Communication Theory:** Terminologies, Elements Of Analog Communication System, Baseband signal, Band-pass signal, Need For Modulation, Electromagnetic Spectrum And Typical Applications, Basics Of Signal (Analog And Digital,) Representation And Analysis (Time and frequency)<br>**Introduction To basics of networking:** Computer network fundamentals, ISO OSI Model: All Layers, TCP/IP Protocol Suite: All Layers, Addressing (Physical, Logical Port and Other), LAN, WAN And MAN, Network Topologies. Guided Media: Twisted-Pair Cable, Coaxial Cable and Fiber-Optic Cable, Unguided Media: Wireless, Radio Waves, Microwaves And Infrared, Wireless frequency spectrum.<br>**Noise:** External Noise, Internal Noise, Noise Calculations, Communication Channel. Discrete and Continuous Channel, Shannon-Hartley Theorem, Channel Capacity, Nyquist and Shanon Theorem, Bandwidth S/N Trade Off | | |
| | **Practical** | | |
| | | | |
| **Unit-II** | **AMPLITUDE AND ANGLE MODULATION** **8 Hours** | | |
| | **Amplitude Modulation**: Amplitude Modulation Techniques (DSBFC, DSBSC, SSB), Generation Of Amplitude Modulated Signals, Frequency Spectrum. | | |

| | |
|---|---|
| | **Angle Modulation Techniques:** Theory Of Angle Modulation Techniques, Practical Issues In Frequency Modulation, Generation Of Frequency Modulation, Frequency Spectrum |
| | **Practical** |
| | |
| **Unit-III** | **PULSE AND DIGITAL MODULATION TECHNIQUES** **8 Hours** |
| | **Pulse Modulation Techniques:** Pulse Analog Modulation Techniques, sampling<br>Pulse Digital Modulation Techniques: PCM, DM, DPCM<br>Average Information, Entropy, Information Rate. Source coding: Shanon-Fano, Huffman and Limpel-Ziv<br>**Digital-to-digital Conversion:** Line Coding, Line Coding Schemes, Block Coding, Scrambling<br>**Digital-to-analog Conversion:** Aspects of Digital-to-Analog Conversion, Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), Phase Shift Keying (PSK), Quadrature Amplitude Modulation (QAM)<br>**Analog-to-analog Conversion:** Amplitude Modulation, Frequency Modulation, Phase Modulation |
| | **Practical** |
| | |
| **Unit-IV** | **ERROR CONTROL CODING AND DATA LINK CONTROL** **8 Hours** |
| | **Error Detection and Correction**:Introduction, Error Detection, Error Correction<br>**Linear Block Codes**: hamming code, Hamming Distance, parity check code<br>**Cyclic Codes**: CRC (Polynomials), Advantages Of Cyclic Codes, Other Cyclic Codes As Examples: CHECKSUM: One's Complement, Internet Checksum<br>**Framing**: fixed-size framing, variable size framing.<br>**Flow control**: flow control protocols.<br>Noiseless channels: simplest protocol, stop-and-wait protocol.<br>Noisy channels: stop-and-wait automatic repeat request, go-back-n automatic repeat request, Selective repeat automatic repeat request, piggybacking |
| | **Practical** |
| | |
| **Unit- V** | **MULTIPLEXING AND MULTIPLE ACCESS** **6 Hours** |
| | **Multiplexing**: FDM, TDM, Synchronous Time-Division Multiplexing, Statistical Time-Division Multiplexing, WDM, Spread Spectrum: FHSS and DSSS<br>**Random access:** ALOHA, CSMA, CSMS/CD and CSMA/ CA<br>**Controlled Access:** Reservation, Polling and Token Passing<br>**Channelization:** FDMA, TDMA and CDMA |
| | **Practical** |
| | |
| **Unit-VI** | **PHYSICAL, MAC LAYER STANDARDS AND SWITCHING** **6 Hours** |
| | LAN hardware: (Switches, routers, hubs, bridges and their types)<br>IEEE 802.3, Fast Ethernet ( MAC Sublayer & Physical Layer ), Gigabit Ethernet ( MAC Sublayer, Physical Layer) Ten-Gigabit Ethernet, Token ring and token bus standards. |

| | | | |
|---|---|---|---|
| | Circuit Switched Networks, Packet (Datagram) Networks, Virtual Circuits, Structure of Circuit and Packet Switches | | |
| | **Practical** | | |
| | | | |
| **Text Books** | **Author** | **Title of Book** | **Publication** |
| T1 | George Kennedy, Brendan Davis, srmPrasanna | Electronic Communication Systems | MGH Education |
| T2 | Behrouz a Forouzan | Data Communications and Networking | McGraw Hill |
| | | | |
| **Reference Books** | | | |
| R1 | Simon Haykin and Michael Moher | Introduction to Analog and Digital Communications | John Wiley & Sons, Inc. |
| R2 | Louis E. Frenzel | Principles Of Electronic Communication Systems (SIE) | Tata McGraw Hill |
| R3 | A S Tanenbaum | Computer Networks | Pearson Education |
| R4 | Roddy &Coolen | Electronic communications | PHI. |
| R5 | Kenedy&DavisTMH | Electronic Communication System | MGH Education |
| | B.P. Lathi | Modern Digital & Analogue Communication Systems | Oxford Press |
| | H. Taub And K.L. Shiling, | Principles of Communication System | Tata Mcgraw Hill Education Private Limited |
| | Irvine | Data Communications and Networks: An Engineering Approach | Wiley |
| | Keshav | An Engineering Approach to Computer Networking | Pearson Education |
| | | | |
| **Self-Learning Facilities** | NPTEL Lecture Series | | |
| | | | |