



**JSS COLLEGE OF ARTS, COMMERCE & SCIENCE
(AUTONOMOUS)**

OOTY ROAD, MYSURU-570 025

(Autonomous under University of Mysore: Re-accredited by NAAC with 'A' Grade)

**Choice Based Credit System and CAGP
MASTERS DEGREE**



Syllabus

2020-22

Postgraduate Department of Computer Science

JSS College of Arts, Commerce and Science

Ooty Road, Mysore-25

2020-2022

PROGRAMME: MASTER OF SCIENCE IN COMPUTER SCIENCE

2 Years /4 Semesters (under CBCS-CAGP)

ADMISSIONS:

- i) 50% seats of the total intake for M.Sc., Computer Science Programme of the College will be filled-up by University of Mysore through Centralized Admission Cell as per University regulations.
- ii) Remaining 50% seats will be filled-up by the College under College Quota.

ADMISSION REQUIREMENT

Eligibility - All students who have 45% (40% for SC/ST) in their B.Sc degree (from Recognised University/ Open University) with Mathematics as one subject and any one-two among Electronics, Instrumentation, Computer Science or Computer Maintenance and Physics as other subject. OR Student with B.Sc (IT) are permitted provided they have not studied Mathematics in their 2 years PUC. OR Diploma (in Electronics, Computer Science and Information Technology) holders with B.Sc (IT) are permitted if they have SSLC level Mathematics. OR BCA degree from recognized University OR equivalent to this from any other recognized University in India or abroad is also permitted.

PROGRAMME OUTCOMES:

After Completing the M.Sc Program Students will be able to:

- PO1. Identify, formulate, and solve computer science problems
- PO2. Design, implement, test, and evaluate a computer system, component, or algorithm to meet desired needs
- PO3. Receive the broad education necessary to understand the impact of computer science solutions in a global and societal context
- PO4. Communicate effectively
- PO5. Success in research or industry related to computer science
- PO6. Have solid knowledge in computer science and engineering, including programming and languages, algorithms, theory, databases, etc.
- PO7. Integrate well into and contribute to the local society and the global community related to computer science
- PO8. Practice high standard of professional ethics
- PO9. Draw on and integrate knowledge from many related areas

PROGRAMME SPECIFIC OUTCOMES:

PSO1. Programmers or the Software Engineers with the sound knowledge of practical and theoretical concepts for developing software.

PSO2. Serve as the Computer Engineers with enhanced knowledge of computers And its building blocks. Work as the Hardware Designers/Engineers with the knowledge of Networking Concepts.

PSO3. Work as the System Engineers and System integrators Serve as the System Administrators with thorough knowledge of DBMS.

PSO4. Work as the Support Engineers and the Technical Writers

PSO5. Work as IT Sales and Marketing person.

PSO6. Serve as the IT Officers in Banks and cooperative societies.

PSO7. Computer Scientist in research and R & D laboratories.

PSO8. Faculty for Graduate and Under graduate Colleges.

MSc. in Computer Science – 2018**I Semester**

Course Code	Course	L:T:P	Credit Value
CSC101	HC1 (Data Structures and Algorithms)	3:0:1	4
CSC102	HC2 (System Software)	2:1:1	4
CSC103	HC3 (Computer Networks)	2:1:1	4
CSC104	SC1		4
CSC105	SC2		4
	TOTAL		20

II Semester

Course Code	Course	L:T:P	Credit Value
CSC201	HC4 (Analysis and Design of Algorithms)	2:1:1	4
CSC202	HC5 (Operating System and Unix)	2:0:2	4
CSC203	HC6 (Computer Graphics)	3:0:1	4
CSC204	SC3		4
CSC205	SC4		4
	TOTAL		20

III Semester

Course Code	Course	L:T:P	Credit Value
CSC301	HC7 (Software Engineering)	3:1:0	4
CSC302	HC8 (Theory of Languages)	3:1:0	4
CSC303	HC9 (Database Management System)	2:1:1	4
CSC304	SC5 / Term Work		4
CSC305	SC6 (Open Elective *****)		4
	TOTAL		20

IV Semester

Course Code	Course	L:T:P	Credit Value
CSC401	HC10 (Major Project)	0:1:7	8
CSC402	SC7		4
CSC403	SC8		4
	TOTAL		16

****** Open Elective Course: III Semester**

Course Code	Course	L:T:P	Credit Value
CSC305	OE- Computer Fundamentals / Programming with C	2:0:2	4

HARD CORE:

Sl. No.	Course	L:T:P	Credit Value
1	Computer Networks	2:1:1	4
2	Computer Graphics	3:0:1	4
3	Data Structures and Algorithms	3:0:1	4
4	Analysis and Design of Algorithms	2:1:1	4
5	DBMS	2:1:1	4
6	Software Engineering	3:1:0	4
7	Theory of Languages	3:0:1	4
8	Operating System and Unix	2:0:2	4
9	System Software	3:1:0	4

SOFT CORE:

Sl. No.	Course	L:T:P	Credit Value
1	Principles of Programming Language and C	2:1:1	4
2	Internet Technology	2:0:2	4
3	Java Programming	2:0:2	4
4	Multimedia	3:1:0	4
5	Microcontroller	3:1:0	4
6	Discrete Mathematics	3:1:0	4
7	Simulation and Modeling	3:1:0	4
8	Operations Research	3:1:0	4
9	Mobile Communication	3:1:0	4
10	C++	2:0:2	4
11	Pattern Recognition	3:0:1	4
12	Image Processing	2:1:1	4
13	Software Testing	3:0:1	4
14	Computational Techniques	3:0:1	4
15	Graph Theory	3:1:0	4
16	OOAD	2:1:1	4

17	Probability and Statistics	3:1:0	4
18	Data Mining	2:1:1	4
19	Artificial Intelligence	3:1:0	4
20	.NET Technologies	2:0:2	4
21	Object Oriented Modeling and Design with UML	2:1:1	4
22	Android Application Development	2:0:2	4
23	Advanced Database Management Systems	2:1:1	4
24	Compiler Design	3:0:1	4

SCHEME OF EXAMINATION AND ASSESMENT:

In view of the CBCS syllabus, Each Course is Assess with Components . Component 1 (C1), Component 2 (C2), and Component 3 (C3),

The following is the scheme which will be followed for the assessment of marks for both theory (HC/ SC/ OE) as well as practicals (HC/ SC) irrespective of the Credits associated with each Course. Thirty percent of the marks will be assessed for the internals (C1 and C2) and remaining seventy percent will be for the semester end examinations (C3). Each Course carries 100 marks and hence thirty marks for internal assessment and remaining seventy marks will be for Semester End Examinations. Out of thirty marks for internals, fifteen marks will be allotted to each C1 and C2 components.

Each theory Course (HC/ SC/ OE) consists of three components namely C1, C2 and C3. C1 and C2 are designated as Internal Assessment (IA) and C3 as Semester End Examination. Each Course (HC/ SC/ OE) carries 100 Marks and hence the allotment of marks to C1, C2 and C3 Components will be fifteen, fifteen and seventy marks respectively. i.e.,

C1 Component : 15 Marks	Internal Assessment Marks
C2 Component : 15 Marks	
C3 Component : 70 Marks	Semester End Examination
Total :	100 Marks

The above will be followed in common for all the theory (HC/ SC/ OE) Courses in all the four semesters.

DATA STRUCTURES & ALGORITHMS

3:0:1

Course Code:CSA100

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

- CO1. Select appropriate data structures as applied to specified problem definition.
- CO2. Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.
- CO3. Implement Linear and Non-Linear data structures.
- CO4. Implement appropriate sorting/searching technique for given problem.
- CO5. Design advance data structure using Non Linear data structure.

UNIT I: Introduction

Introduction to Stages in Problem Solving, Difference between Data Type, Data Structure and Storage Structure, Formal Definition of Data Structure, Classification of Data Structure
Analysis of Algorithms: Algorithm, Time Complexity and Space Complexity, O-Notation, Omega Notation and Theta Notation.
Primitive Data Structure: Integer, Real, Character and Boolean and Its Storage Representation

UNIT II: Non-Primitive Data Structures

Arrays: Introduction to Array Data Structure and Its Representation, Sparse Matrix Representation. Introduction to Structures
Stacks: Introduction, Implementation Using Arrays, Applications - Tower Of Hanoi, Expression Evaluation, Conversion of Expressions

UNIT III: Stacks and Queues

Queue: Introduction, Types – Ordinary, Circular, Doubly Ended, Priority, Implementation Using Arrays

Linked List: Introduction, Types, Operations, Implementation of Stacks and Queues Using Linked List

UNIT IV: Non-Linear Data Structures

Graphs: Introduction, Basic Terminologies, Graph Representation – Adjacency and Incidence Matrix Representation

Trees: Introduction, Binary Tree Representation – Array and Linked List Representations, Traversals – Preorder, In-order, Post order, Binary Search Tree, Introduction to B-Trees

Hash Tables: Direct Address Tables, Hash Tables, Hash Functions, Open and Closed Addressing

References:

1. Data Structures and Algorithms, 2nd Edition, 2006 , Andrew Tanenbaum
2. “An Introduction to Data Structures, with Applications” McGraw Hill, Kongakusha 1976, Trembley and Sorenson
3. “Data Structures” SBCS Publication, 1980, Horowitz and Sahni
4. Data Structures using c, A K Sharma
5. Data Structures and program design in C, Kruse Robert L
6. Data Structures and analysis in C, Mark Allen Weiss
7. Data Structures and Algorithms, Alfred V AHO and Jeffrey D Ullman

SYSTEM SOFTWARE

2:1:1

Course Code:CSA110

Course Outcomes:

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

CO1. Understand fundamentals of language processing and grammar

CO2. Apply knowledge of compilation and code optimization steps to mimic a simple compiler

CO3. Demonstrate the working of various system software like assembler, loader, linker, editor and device driver

UNIT 1

Background: Machine Structure, Evolution of the components of a programming system, evolution of operating system, operating system user view point functions, facilities

General Machine Structure, General Approach to a new machine, Machine Structure – 360 and 370, Assembly Language

UNIT 2

Assemblers: General design procedure, design of an assembler.

Macro language and macro processor, macro instructions, features of macro, implementation

UNIT 3

Loaders, different types of loaders, loader schemes, design of an absolute loader, design of direct linking loader.

Compilers: Structure and phases

UNIT 4

Lex and yacc: The Simplest lex Program, Recognizing Words with Lex, Parser-Lexer

Communication, Regular Expressions, Grammars, Shift/Reduce Parsing, Structure of lex and yacc Programs, Programs in lex and yacc

References:

1. Systems Programming by Donovan
2. Principles of Compiler design by Ullman
3. System programming by Dhamdhare
4. Lex and yacc by John R Levine, Tony Mason and Doug Brown
5. System Software- Prof. Liland L Beck.

6. System Software- Prof. John R Levine

COMPUTER NETWORKS

2:1:1

Course code:CSA120

Course Outcomes:

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

CO1. Master the terminology and concepts of the OSI reference model and the TCP-IP reference model.

CO2. Study the basic taxonomy and terminology of the computer networking and enumerate the layers of OSI model and TCP/IP model.

CO3. Master the concepts of protocols, network interfaces, and design/performance issues in local area networks and wide area networks

CO4. Acquire knowledge of Application layer and Presentation layer paradigms and protocols.

CO5. Study Session layer design issues, Transport layer services, and protocols.

CO6. Gain core knowledge of Network layer routing protocols and IP addressing.

CO7. Study data link layer concepts, design issues, and protocols.

CO8. Read the fundamentals and basics of Physical layer, and will apply them in real time applications.

CO9. Familiar with wireless networking concepts

CO10. Familiar with contemporary issues in networking technologies

CO11. Familiar with network tools and network programming

UNIT 1

USES OF COMPUTER NETWORK: Business Applications, Home Applications, Mobile Users, Social Issues

NETWORK HARDWARE: Local Area Networks, Metropolitan Area Networks, Wide Area Networks, Wireless Networks, Home Networks, Internetworks

NETWORK SOFTWARE: Protocol Hierarchies, Design Issues for the Layers, Connection-Oriented and Connectionless Service, Service Primitives, the Relationship of Services to Protocols

REFERENCE MODELS: The OSI Reference Model, The TCP/IP Reference Model, A comparison of OSI and TCP/IP Reference Model

UNIT 2

THE PHYSICAL LAYER

THE THEORETICAL BASIS FOR DATA COMMUNICATION: Fourier Analysis, Bandwidth-Limited Signals, The Maximum Data Rate of a Channel

GUIDED TRANSMISSION MEDIA: Magnetic Media, Twisted Pairs, Coaxial Cable, Fiber Optics

WIRELESS TRANSMISSION: The Electromagnetic Spectrum, Radio Transmission, Microwave Transmission, Infrared and Millimeter Waves, Light wave Transmission

COMMUNICATION SATELLITES: Geostationary Satellites, Medium-Earth Orbit Satellites, Low-Earth Orbit Satellites, Satellites versus Fiber

THE DATA LINK LAYER

DATA LINK LAYER DESIGN ISSUES: Services Provided to the Network Layer, Framing, Error Control, Flow Control

ERROR DETECTION AND CORRECTION: Error-Correcting Codes, Error-Detecting Codes

ELEMENTARY DATA LINK PROTOCOLS: A Utopian Simplex Protocol, A Simplex Stop-and-Wait Protocol

SLIDING WINDOW PROTOCOLS: A One-Bit Sliding Window Protocol, A Protocol Using Go-Back-N

UNIT 3

THE MEDIUM ACCESS CONTROL SUBLAYER

THE CHANNEL ALLOCATION PROBLEM: Static Channel Allocation, Dynamic Channel Allocation

MULTIPLE ACCESS PROTOCOLS: ALOHA, Carrier Sense Multiple Access Protocols, Collision-Free Protocols, Limited-Contention Protocols

ETHERNET: Ethernet Cabling, Manchester Encoding, The Ethernet MAC sub layer protocol, the binary exponential back off algorithm, Ethernet Performance, Switched Ethernet, IEEE 802.2: Logical Link Control

THE NETWORK LAYER

NETWORK LAYER DESIGN ISSUES: Store-and-Forward Packet Switching, Services Provided to the Transport Layer, Implementation of Connectionless Service, Implementation of Connection-Oriented Service, Comparison of Virtual-Circuit and Datagram Networks

ROUTING ALGORITHMS: The Optimality Principle, Shortest Path Algorithm, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing

CONGESTION CONTROL ALGORITHMS: Approaches to Congestion Control, Traffic-Aware Routing, Admission Control, Traffic Throttling, Traffic Shaping

INTERNETWORKING: How Networks Differ, How Networks Can Be Connected, Tunneling, Internetwork Routing, Fragmentation

THE NETWORK LAYER IN THE INTERNET: The IP Protocol, IP Addresses, IPv6

UNIT 4

THE TRANSPORT LAYER

THE TRANSPORT SERVICE: Services Provided to the Upper Layers, Transport Service Primitives

ELEMENTS OF TRANSPORT PROTOCOLS: Addressing, Connection Establishment, Connection Release, Error Control and Flow Control, Multiplexing, Crash Recovery

THE INTERNET TRANSPORT PROTOCOLS: UDP: Introduction to UDP, Remote Procedure Call, Real-Time Transport Protocols

THE INTERNET TRANSPORT PROTOCOLS: TCP: Introduction to TCP, The TCP Service Model, The TCP Protocol, The TCP Segment Header, TCP Connection Establishment, TCP Connection Release, TCP Connection Management Modeling, TCP Sliding Window, TCP Timer Management

THE APPLICATION LAYER

DNS—THE DOMAIN NAME SYSTEM: The DNS Name Space, Resource Records, Name Servers

ELECTRONIC MAIL: Architecture and Services, The User Agent, Message Formats, Message Transfer, Final Delivery

NETWORK SECURITY:

CRYPTOGRAPHY: Introduction to Cryptography, Substitution Ciphers, Transposition Ciphers, Two Fundamental Cryptographic Principles

SYMMETRIC-KEY ALGORITHMS: DES—The Data Encryption Standard, Cipher Modes

PUBLIC-KEY ALGORITHMS: RSA, Other Public-Key Algorithms

DIGITAL SIGNATURES: Symmetric-Key Signatures, Public-Key Signatures

MANAGEMENT OF PUBLIC KEYS: Certificates

Reference Books:

1. Computer Networks, 5th Edition, Prentice Hall, 2006, Andrew S. Tanenbaum & David J. Wetherall
2. Data & Computer Communications, 6th Edition, Pearson Education, 2002, William Stallings
3. Computer Networks: 3rd Edition, Elsevier, 2003, Larry L. Peterson & Bruce S. Davie
4. Data Communication & Networking, 4th Edition, Mc Graw Hill, 2006, Behrouza Forouzan
5. Computer & Networks with Internet Applications, 4th Edition, Pearson Education, 2004, Douglas E. Comer

ANALYSIS AND DESIGN OF ALGORITHMS

2:1:1

Course Code:CSB060

Course Outcomes:

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

- CO1. Analyze different scenarios for running time of algorithms using asymptotic notations and Design using Recursion.
- CO2. Apply divide and conquer strategy for design of various algorithms.
- CO3. Develop algorithms for well known problems using greedy methods.
- CO4. Describe and apply dynamic-programming approach for designing graph and matrix based algorithms.
- CO5. Understand the concept of backtracking for traversal and search algorithms.
- CO6. Apply the knowledge earned to determine the efficiency of algorithms considering time and space tradeoffs.

UNIT 1

INTRODUCTION: Algorithm specification, pseudo code conventions

PERFORMANCE ANALYSIS: Space Complexity, Time Complexity, Asymptotic Notation, Mathematical Analysis: Recursive and Non recursive algorithms

BRUTE FORCE – Bubble Sort, Selection Sort, Sequential Search, String Matching

UNIT 2

DIVIDE- AND – CONQUER: General Method, Binary Search, Finding the Maximum and Minimum, Merge Sort, Quick Sort, Strassen's Matrix Multiplication

THE GREEDY METHOD: The General Method, Knapsack Problem, Tree Vertex Splitting, Job Sequencing with Deadlines, Minimum-Cost Spanning Trees - Prim's Algorithm, Kruskal's Algorithm, Optimal Storage on Tapes, Optimal Merge Patterns, Single-Source Shortest Paths.

UNIT 3

DYNAMIC PROGRAMMING: The General Method, Binomial Coefficient, Multistage Graphs, All Pairs Shortest Paths Single-Source Shortest Paths: General Weights, String Editing, 0/1 Knapsack, the Traveling Salesperson Problem

BACKTRACKING: The General Method, the 8-Queens Problem, Sum of Subsets, Graph Coloring, Hamiltonian Cycles

UNIT 4

Elementary Graph Problems: Depth First Search, Breadth First Search, Topological Sort

NP-Hard and NP-Complete Problems: Basic Concepts, Nondeterministic Algorithms, The Classes NP-Hard And NP-Complete

NP-Hard Graph Problems: Clique Decision Problem (CDP), Node Cover Decision Problem, Chromatic Number Decision Problem (CNDP), Traveling Salesperson Decision Problem (TSP), AND/OR Graph Decision Problem (AOG)

References:

1. Analysis and Design of Algorithms: Horowitz Sahani
2. Analysis and Design of algorithms: Trembly
3. Introduction to Algorithms: Thomas H. Cormen
4. Analysis and Design of Algorithm: Padma Reddy
4. Introduction to the design and analysis of algorithm: Anany Levitin.
5. Design and analysis of algorithm: S Shridhar.

OPERATING SYSTEM and UNIX

2:0:2

Course Code:CSB070

Course Outcomes:

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

- CO1. Understand device drivers
- CO2. Write applications with improved performance and stability
- CO3. Write set of small commands and utilities that do specific tasks well
- CO4. Run multiple programs each at the same time without interfering with each other or crashing the system.
- CO5. Implement Commands of UNIX.
- CO6. Implement various file processing commands and shell Programming.

UNIT 1

Introduction to Operating System, Operating System Structures

Process Management: Processes, CPU Scheduling

UNIT 2

Deadlocks, Storage Management: Memory management, Virtual Memory, File-System Interface

UNIT 3

Introduction: Why UNIX? The Unix Environment, UNIX Structure, accessing UNIX, UNIX commands

File Systems: File Names, File Types, Regular Files, Directories, File System Implementation, Operations unique to directories, Operations unique to regular files, Operations common to both.

Vi editor, local commands, range commands in vi, global commands in vi, rearrange text in vi, ex editor.

UNIT 4

Introduction to shells: Unix Session, Standard Streams, Redirection, pipes, tee command, command execution, command line editing, quotes, command substitution, job control, aliases, variables, predefined variables, options, shell/environment customization.

Security and file permission: User and groups, Security levels, changing permissions, User masks, Changing ownership and group, Regular expressions: Atoms, Operators, grep: operation, grep family, Searching for file content, sed and awk.

C Shell Programming: Basic script concepts, Expressions, Decision making selections,, Built in commands, Scripting techniques, Shell environment & Script, Script examples

References:

1. Unix and Shell Programming, Behrouz A Forouzan and Richard F.Gilberg, 2nd Edition, 2003, Thomson.
2. System Programming and Operating Systems, Dhamdhare. D.M., 4th Edition, TataMcGraw Hill, 2006
3. A Practical Guide to Linux, Mark G. Sobell, 1st Edition, 2002, Pearson Education (Chapters:1 to 5, 8, 10, 11, 15)
4. UNIX: The Complete Reference, Kenneth Rosen and others, 2nd Edition, 2002,Obsborne/McGraw Hill
5. Design of the UNIX Operating System, Maurice J Bach.
6. Operating System: A Modern Perspective , Gary J Nutt.

COMPUTER GRAPHICS

3:0:1

Course code:CSB080

Course Outcomes:

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

CO1. Utilize the components of a graphics system and become familiar with building approach of graphics system components and algorithms related with them.

CO 2. Learn the basic principles of 3- dimensional computer graphics.

CO3. Provide an understanding of how to scan convert the basic geometrical primitives, how to transform the shapes to fit them as per the picture definition.

CO4. Provide an understanding of mapping from a world coordinates to device coordinates, clipping, and projections.

CO 5. Implement the applications of computer graphics concepts in the development of computer games, information visualization, and business applications.

CO6. Comprehend and analyze the fundamentals of animation, virtual reality, underlying technologies, principles

UNIT 1

Introduction, Video Display Devices, Refresh Cathode-Ray Tubes, Raster-Scan Display, Random-Scan Displays, Color CRT Monitors, Flat-Panel Displays, Raster Scan Systems, Input devices.

Output primitives: Points & Lines, Line Drawing Algorithms, Loading the Frame Buffer, Circle Generating Algorithms, Pixel Addressing and Object Geometry

UNIT 2

Two dimensional transformations Basic & other transformations, Matrix representations, Homogeneous coordinates Composite transformations, General-pivot-point transformations.

Three Dimensional Transformations: Introduction to 3D Translation, Rotation & Scaling, Other Transformations, Modeling and Co-ordinate Transformations.

UNIT 3

Three Dimensional Viewing :Viewing Pipeling, window to viewport transformations, Projections, Types of projections.

Graphical User Interface & Interactive Input Methods : The User Dialogue, Windows & Icons, feedback, Input of Graphical Data, Interactive Picture Construction Techniques, Basic Positioning Methods, Constraints, Grids, Gravity Field, Rubber-Band Methods, Dragging, Painting & Drawing

UNIT 4

Curves & Surfaces: Properties, Bezier curves properties, Design techniques, Bezier surfaces, Displaying curves & surfaces

Hidden line removal algorithms

Introduction to fractals, Serpinsky's triangle, Construction, Koch curves.

Windowing & Clipping: Clipping operations, Line clipping algorithms, point clipping, text clipping, polygon clipping algorithms, Exterior clipping

Reference:

1. "Computer Graphics", Pearson Education, Donald D. Hern and M. Pauline Baker
2. "Principles of Interactive Computer Graphics" McGraw Hill 1989, W. M. Newman and Robert Sproull
3. "Computer Graphics a Programming Approach" McGraw Hill 1987, Steven Harrington
4. "Schaums outline of theory and problems of Computer Graphics" 2nd printing 1987, 1986 Edition, Roy A Plastock and Gardon Kelley
5. "Procedural Elements of Computer Graphics" McGraw Hill 2nd edition 1990, David F Frogers and J Alan Adams
6. Computer Graphics, James.D.Foley, A Vandam etal

SOFTWARE ENGINEERING

3:1:0

Course Code:CSC040

Course Outcomes:

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

CO1. Understand the nature of software development and software life cycle process models, agile software development, SCRUM and other agile practices.

CO2 Learn methods of capturing, specifying, visualizing and analyzing software requirements.

CO3. Understand concepts and principles of software design and user-centric approach and principles of effective user interfaces.

CO 4. Basics of testing and understanding concept of software quality assurance and software configuration management process.

CO 5. Understand need of project management and project management life cycle.

CO 6. Understand project scheduling concept and risk management associated to various types of projects.

UNIT 1

INTRODUCTION TO SOFTWARE ENGINEERING: Software and Software Engineering, phases in Software Development, Software Development Process models, Agile modeling, Introduction to metrics.

UNIT 2

SOFTWARE REQUIREMENT SPECIFICATION: Role of SRS, Problem Analysis, Requirement specification, validation, metrics, Monitoring and control

PLANNING A SOFTWARE PROJECT: Cost Estimation, Project Scheduling, Staffing personal planning, Team Structure, Software Configuration Management, Quality Assurance Plans, Project Monitoring plans, Risk Management.

UNIT 3

SYSTEM DESIGN: Design Objectives, design principles, Module level Concepts, Design methodology – object oriented approach Design Specification, Verification, Metrics, monitoring and control.

DETAILED DESIGN: Module Specification, Detailed design and process design Language, Verification

UNIT 4

CODING: Programming practice, Verification, Metrics

TESTING: Testing Fundamentals, Fundamental testing, Structural Testing. Testing process
Clean Room approach

References:

1. “An Integrated approach to the Software Engineering” 2ed. Narosa Publishing House, New Delhi, 2002, Pankaj Jalote
2. Software Engineering Principles & Practice - 3rd Edition, Tata Mc Graw Hill Companies – 2006, Waman S Jawadekar
3. Software Engineering A Practitioner’s Approach - 6th Edition McGraw Hill – 2005, Roger S Pressman
4. Software Engineering - 7th Edition : Pearson Education Ltd- 2006, Sommerville

THEORY OF LANGUAGES

3:1:0

Course Code:CSC050

Course Outcomes:

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

CO1. Design different types of Finite Automata and Machines as Acceptor, Verifier and Translator.

CO2. Understand, design, analyze and interpret Context Free languages, Expression and Grammars.

CO3. Design different types of Push down Automata as Simple Parser.

CO4. Design different types of Turing Machines as Acceptor, Verifier, Translator and Basic computing machine.

UNIT - 1

INTRODUCTION TO FINITE AUTOMATA: Introduction to Finite Automata; The central concepts of Automata theory; Deterministic finite automata; Nondeterministic finite automata.

FINITE AUTOMATA, REGULAR EXPRESSIONS: An application of finite automata; Finite automata with Epsilon-transitions; Regular expressions; Finite Automata and Regular Expressions

UNIT - 2

REGULAR LANGUAGES, PROPERTIES OF REGULAR LANGUAGES: Regular languages; Proving languages not to be regular languages; Closure properties of regular languages; Equivalence and minimization of automata.

CONTEXT-FREE GRAMMARS AND LANGUAGES: Context –free grammars; Parse trees; Ambiguity in grammars and Languages.

UNIT – 3

PUSHDOWN AUTOMATA: Definition of the Pushdown automata; the languages of a PDA; Equivalence of PDA's and CFG's; Deterministic Pushdown Automata.

PROPERTIES OF CONTEXT-FREE LANGUAGES: Normal forms for CFGs; The pumping lemma for CFGs; Closure properties of CFL

UNIT - 4

TURING MACHINE: The turing machine; Extensions to the basic Turing Machines;

UNDECIDABILITY: A Language that is not recursively enumerable; An Undecidable problem that is RE; Post's Correspondence problem

REFERENCES:

1. Introduction to Automata Theory, Languages and Computation – John E.. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman:, 3rd Edition, Pearson education, 2007.
2. Fundamentals of the Theory of Computation: Principles and Practice – Raymond Greenlaw, H.James Hoove, Morgan Kaufmann, 1998.
3. Introduction to Languages and Automata Theory – John C Martin, 3rd Edition, Tata McGraw-Hill, 2007.

DATABASE MANAGEMENT SYSTEM

2:1:1

Course Code:CSC060

Course Outcomes:

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

- CO1. Explain the features of database management systems and Relational database.
- CO2. Design conceptual models of a database using ER modelling for real life applications and also construct queries in Relational Algebra.
- CO3. Create and populate a RDBMS for a real life application, with constraints and keys, using SQL.
- CO4. Retrieve any type of information from a data base by formulating complex queries in SQL.
- CO5. Analyze the existing design of a database schema and apply concepts of normalization to design an optimal database.
- CO6. Build indexing mechanisms for efficient retrieval of information from a database

UNIT 1

Introduction and conceptual modeling databases and Database users, Data modeling using the entity relationship (ER) model, the enhanced entity – relationship (EER) model.

UNIT 2

Relational model: Concepts constraints, Languages, Design and programming.

The relational data model and relational database constraints, Relational algebra and relational calculus, Introduction to SQL Programming technique

UNIT 3

Database design theory and methodology functional dependencies and Normalization for relational database, Relational database design algorithms and further dependencies, practical database design methodology and use of UML diagrams.

UNIT 4

Introduction to transaction processing concepts and theory recovery

REFERENCES

1. Fundamentals of database system – 5th Edition – Ramez elmasri, Navathe – Person edition
- 2 .An introduction to database system – 8th Edition – C. J. Date, Kannan – Person Education
- 3.Database system concepts – 5th Edition – Korth, Sudarshan – McGraw Bill Edition

4. Database Management System- Raghuramakrishnan.
5. An Introduction to Database System- Bipin Desai
6. Principles of Database System- J D Ullman

Softcore:

PRINCIPLES OF PROGRAMMING LANGUAGES AND 'C'

2:1:1

Course Code:CSA020

Course Outcomes:

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

- CO1. Analyzing semantic issues associated with function implementations, including variable binding, scoping rules, parameter passing, and exception handling.
- CO2. Implementation techniques for interpreted functional languages.
- CO3. Using object-oriented languages.
- CO4. Familiar with design issues of object-oriented and functional languages.
- CO5. Familiar with language abstraction constructs of classes, interfaces, packages, and procedures.
- CO6. Familiar with implementation of object-oriented languages.
- CO7. Familiar with using functional languages

UNIT 1: Preliminaries

Reasons for studying concepts of programming languages, Programming domains, Language evaluation criteria, Implementation methods Names, Bindings, Type Checking, and Scopes Introduction, Names, Variables, The concept of Binding, Type Checking, String Checking, Type Compatibility, Scope, Scope and Lifetime, Referencing Environments, Named Constants, Variable Initialization.

UNIT 2: Data Types

Introduction, Primitive Data types, Character String Types, User-Defined Ordinal Types, Array Types and Associative Arrays, Record Types, Union Types, Set Types, Pointes Types.

Expression, Assignment Statements and Statement Level Control Structures

Introduction, Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational And Boolean Expressions, Short-Circuit Evaluation, Assignment Statements, Mixed-Mode assignment.

UNIT 3

Compound Statements, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands, Conclusion.

Subprograms: Introduction, fundamentals, design issues, local referencing environments, parameter passing methods, overloaded programs, generic subprograms, coroutines, user defined overloaded operators

UNIT 4

C LANGUAGE: C Fundamentals, Operation data input and output, Control statements, Function Storage classes, Arrays, Pointers, structures and unions, Enumeration, Command line parameters, Macros, 'C' processor.

Reference:

1. Concepts of Programming Languages, Eight Edition, Pearson, Robert W. Sebesta
2. Foundation for Programming Languages, John C Mitchell
3. Principles of Programming Language, Chopra Rajiv
4. Principles of Programming Language, Dowek
5. Types and Programming Languages, Benjamin C Pierce
6. Programming Languages: Principle and Practices ,3rd Edition, Kenneth C Louden

INTERNET TECHNOLOGY

2:0:2

Course Code:CSD220

Course Outcomes:

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

- CO1. Develop analytical ability in network technology.
- CO2. Create quality websites
- CO3. Work individually as a web designer and set up their own business
- CO4. Get the job opportunities in most companies for professional web designers and build websites more visually elegant and interactive
- CO5. Implement interactive web page(s) using HTML, CSS and JavaScript.
- CO6. Design a responsive web site using HTML5 and CSS3.

UNIT 1

Fundamentals: introduction to the Internet, WWW, Web Browsers, Web Servers, URL, Multipurpose Internet Mail Extensions (MIME), HTTP, Security, Introduction to HTML: Origins and Evolution, Basic Syntax, Document Structure, Basic tags, Images, Links, Lists, Tables, Forms, Frames.

UNIT 2

Introduction to XML: Syntax of XML, XML Document Structure, Document Type Definition.

Introduction to XHTML: Origins and Evolution, Basic Syntax, Document Structure, Basic tags, Images, Links, Lists, Tables, Forms, Frames, Syntactic difference between HTML and XHTML.

Cascading Style Sheets (CSS): Introduction, levels of style sheets, Selector Forms, Property value forms, Font properties, Color, Alignment of Text, Box model, Background Images, and <div> tags.

UNIT 3

The basics of JavaScript: Overview, Object Orientation and JavaScript, General syntactic characteristics, Primitives, Operations and Expressions, Screen Output and Keyboard Input, Control Statements, Objects, Arrays, Functions, Constructors, Errors.

JavaScript and XHTML Documents: Element access, Events and Event Handling, Handling Events from Body elements, Handling Events from Button elements, Handling Events from Text Box and password elements.

Dynamic Documents with JavaScript: Introduction, Positioning Elements, Moving Elements, Element Visibility, Changing Colors and Fonts, Dynamic Content, Stacking Elements, Locating Mouse Cursor, Reacting to Mouse Click, Slow Movement of Elements, Dragging and Dropping Elements.

UNIT 4

Introduction to PHP: Origins and Uses, Overview, General Syntactic Characteristics, Primitive, Operations and Expressions, Output, Control Statements, Arrays, Functions, Form Handling, Cookies.

REFERENCES:

1. Programming the World Wide Web – by Robert W. Sabesta 4th Edition Pearson Publications
2. HTML and XHTML the Complete Reference.
3. How to program the World Wide Web – by Deitel and Deitel
4. Mastering in HTML – by Ray and Ray.
5. Web programming and Internet Technologies: An E Commerce approach- By Porter ` Seobey and Pawan Lingras.
6. Internet Technology and Information services by Joseph Miller

JAVA Programming

2:0:2

Course Code:CSA270

Course Outcomes:

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

- C01. Understand concept of Object Oriented Programming & Java Programming Constructs.
- C02. Understand basic concepts of Java such as operators, classes, objects, inheritance, packages ,Enumeration and various keywords.
- C03. Understand the concept of exception handling and Input/Output operations.
- C04. Design the applications of Java & Java applet.
- C05. Analyze & Design the concept of Event Handling and Abstract Window Toolkit.

UNIT 1

History and evolution of Java, An overview of Java, Data types, variables and arrays, Operators, Control statements- Introducing classes ,A closer look at methods and classes, Inheritance, Packages and interfaces.

UNIT 2

Exception handling, Multithreaded Programming, Enumeration, Autotoxins, I/O, Applets

UNIT 3

Networking, Event handling, Swings.

UNIT 4

String handling, Collection framework, Introduction to J2EE, Java servlet, Java server pages (JSP) and HTML, JDBC objects.

REFERENCES:

- 1. The complete reference Java – 7th Edition – Herbert Schildt – Tata Mcgraw hill Edition.
- 2. The complete reference J2EE – Jem Keogh – Tata Mcgraw hill Edition.
- 3. Object Oriented Programming with Java- M T Somashekara, D S Guru and K S Manjunatha.
- 4. The Complete Reference 7th Edition Herbert Schiidt
- 5. Introduction to Java Programming – E Balaguruswamy
- 6. Head First Java – 2nd Edition
- 7. Core Java- Horst Mann, C S –8th Edition-Cornell.

8. Core Servlet and Java Server pages- Hall, M-Brown L

MULTIMEDIA

3:1:0

Course Code:CSD210

Course Outcomes:

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

- CO1. Understand various file formats for audio, video and text media.
- CO2. Develop various Multimedia Systems applicable in real time.
- CO3. Design interactive multimedia software.
- CO4. Apply various networking protocols for multimedia applications.
- CO5. Develop understanding of technical aspect of Multimedia Systems

UNIT 1

Introduction to Multimedia (MM) Communication, Scope, Range, Feasibility and Challenges of MM Communication Key aspects of MM: Compression, Coding, Transmission and Replay.

UNIT 2

Types of Compression: Quantization, Coding as PCM, DPCM, ADPCM. Simple Encoder and Decoders based on PCM Samples. Introduction to Transform domain Compression. Introduction to Audio part of MPEG, Psychoacoustics

UNIT 3

Compression in Spatial Domain Algorithms for Data Compression in Transform Domain: DCT. Variable Length Coding, Huffman code. Variable Length Coding: Arithmetic Coding. Introduction to JPEG 2000 Standard, Encoders-Decoders based on this. Audio Compression and MPEG Audio

UNIT 4

Fundamental concepts of Video. MPEG Architecture Details: Audio-Video- Systems. Video Coding standard related to H.263 and H. 264. MPEG- 1, 2 Video. MPEG- 4 : Video. Streaming and Transport Issues: Multiplexing, Synchronization and File formats. Errors in MPEG and Error handling, Concealment. Buffer structures and Buffer Management

Introduction to MPEG-7 and MPEG-21., HDTV. Content based Image Retrievals and Digital Libraries.

References:

1. Fundamentals of Multimedia, 2nd Ed, Pearson, 2005, Ze-Nian Li and Mark Drew
2. Multimedia Communications., Pearson, 2005, Fred Halsall
3. Introduction to Data Compression, 3rd Ed, Morgan Kaufman (India Ed), 2005, Khalid Sayood
4. The DATA compression; The Complete Reference, 3rd Ed, Springer (India Ed), 2006, David Solomon
5. Multimedia foundations: A Core Concepts of for Digital Design, Vic Cost Ello
6. Multimedia: Making it work, 9th edition, Tay Vaughan

MICROCONTROLLERS

3:1:0

Course Code:CSC210

Course Outcomes:

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

- CO1. Describe the architecture of 8051 microcontroller and write embedded program for 8051 microcontroller.
- CO2. Design the interfacing for 8051 microcontroller.
- CO3. Understand the concepts of ARM architecture.
- CO4. Demonstrate the open source RTOS and solve the design issues for the same.
- CO5. Select elements for an embedded systems tool.
- CO6. Understand the concept and architecture of embedded systems

UNIT 1

Introduction, Numbering system and binary arithmetic,

UNIT 2

The 8051 Architecture, Basic assembly language programming concepts, Moving data,

UNIT 3

Logical operations, Arithmetic operations Jump and call instructions

UNIT 4

An 8051 Microcontroller design, Applications

References

1. “The 8051 Microcontroller”, 3rd Edition, Thomson India edition, 2007, Kenneth Ayala
2. “Programming and customizing the 8051 microcontroller”, Tata McGraw-Hill edition, 2006, Myke prick
3. “The 8051 Microcontroller and embedded systems”, Pearson India, 2006, Muhammad Ali Mazidi & Janice Gillispie Mazidi,
4. Microcontroller and Embedded System, Mazidi, M A- Mazidi
5. Microcontroller: Architecture, Programming and application, Ayala, Kenneth

DISCRETE MATHEMATICS

3:1:0

Course Code:CSA260

Course Outcomes:

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

- CO1. Construct simple mathematical proofs and possess the ability to verify them.
- CO2. Have substantial experience to comprehend formal logical arguments .
- CO3. skillfull in expressing mathematical properties formally via the formal language of propositional logic and predicate logic.
- CO4. Specify and manipulate basic mathematical objects such as sets, functions, and relations and will also be able to verify simple mathematical properties that these objects possess.
- CO5. Apply basic counting techniques to solve combinatorial problems .
- CO6. Use various techniques of mathematical induction (weak, strong and structural induction) to prove simple mathematical properties of a variety of discrete structures

UNIT 1

Set Theory: Sets and Subsets. Operations on sets, Countable and uncountable sets, The addition principal, the concept of probability.

Mathematical Logic: Propositions, Logical Connectives, Tautologies; Contradictions, Logical equivalence, Application to switching networks, Duality, Commentates NAND and NOR, Converse, Inverse and Contrapositive, Rules of inference.

UNIT 2

Open statements; Quantifiers, Logical Implication involving Quantifiers, Statements with more than one variable, Methods of proof and disproof, Mathematical Induction.

UNIT 3

Relations and Ordering: Cartesian products of sets, Relations, Paths in relations and digraphs, Operations on relations, Composition of relations, Properties of relations, Equivalence relations, Partial orders, Total Orders, External elements in posets, Lattices.

Functions: Functions, Types of functions, Composition of function, Invertible functions, Permutation Function.

UNIT 4

Fundamental principles of counting: Principles of inclusion and exclusion: The rule of sum and product, Permutations, combinations: The binomial theorem, combinations with repetition,

Ramsey number, the Catalan numbers, sterling number and bell numbers, Generalizations of principles, the pigeonhole principle, Derangements-Nothing is in its Right place, Rook polynomials, Arrangements with Forbidden positions.

References:

1. Discrete Mathematics by Dr. Chandrashekar S .
2. Discrete and combinational Mathematics by Ralph P. Grimaldi, 5th edition, Addison Wesley, 2004
3. Discrete mathematical structures by Kolman, Robert C Busby and Sharon., 6th Edition, Prentice Hall, 2008
4. Discrete Mathematics and Application by Kenneth H Rosen.
5. Discrete Mathematics by Norman L Biggs.

SIMULATION & MODELLING

3:1:0

Course Code:CS/A/B/C/D300

Course Outcomes:

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

- CO1. Understand the definition of simulation and how to develop and analyze a simulation model
- CO2. Understand the fundamental logic, structure, components and management of simulation modeling
- CO3. Demonstrate knowledge of how to use Arena
- CO4. Build a simulation model with basic operations and inputs
- CO5. Build a simulation model with detailed operations
- CO6. Perform statistical analysis of output from terminating simulation

UNIT 1

Introduction, Simulation of prepursuit problem, A system & its model, Simulation of an inventory problem, The basic nature of simulation

Simulation of continuous systems: A chemical reactor, Numerical integration vs continuous system simulation, Selection of an integration formula, Runge Kutta integration formulas, simulation of a servo system, Simulation of a water reservoir system, Analog vs digital simulation

UNIT 2

Discrete system simulation

Fixed time-step vs event to event model, On simulating randomness, Generation of random numbers, Generation of non uniformly distributed random numbers, Monte Carlo computation vs stochastic simulation

UNIT 3

Simulation of queuing systems

Rudiments of queuing theory, simulation of single server queue, Simulation of two server queue, Simulation more general queues.

Simulation of PERT network

UNIT 4

Network model of a project, Analysis of an activity network, Critical path computation, Uncertainties in activity duration, Simulation of an activity network, Computer program for simulation, Resource allocation and cost considerations, Inventory control & forecasting Elements of inventory theory, More complex inventory models, simulation examples, Generation of Poisson & erlang variates, Forecasting & regression analysis
Design and Evaluation of Simulation Experiments
Length of simulation runs, Variance reduction techniques, Experimental layout, validation

References:

1. System Simulation with Digital Computer Narsingh Deo
2. System Simulation and Modeling - Sengutta
3. Computer Methods for solving Dynamic Separation problems- C D Holland and A I Liapis.
4. Fundamentals of Modeling separation Process- C D Holand.
5. Process Modeling-M M Denn

OPERATIONS RESEARCH

3:1:0

Course Code: CS A/B/C/D 310

Course Outcomes:

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

CO1: Formulate and solve mathematical model (linear programming problem) for a physical situations like production, distribution of goods and economics.

CO2: Apply the concept of simplex method and its extensions to dual simplex algorithm.

CO3: Solve the problem of transporting the products from origins to destinations with least transportation cost.

CO4: Convert and solve the practical situations into non-linear programming problem.

CO5: Identify the resources required for a project and generate a plan and work schedule

UNIT 1

Introduction: formulation of LP problems, graphical solution of LP problems, General formulation of L P problems, Slack & Surplus variables, Standard form, Matrix form, Simplex method, Revised Simplex method, Dual simplex

UNIT 2

Assignment model, Transportation model, Game theory

Probability: Introduction, Basic terms of probability, The Addition law of probability, discrete & continuous, variables, random variables, probability distribution of random variables, Mean variance& standard deviation, Mathematical expectation of a random variable.

UNIT 3

Queuing theory

Introduction, queuing system, distribution, Kendall's Notation, Classification, model I (m/m/1).

UNIT 4

Project management by PERT CPM

Introduction, history, Applications, Basic steps, Network diagram representation, rules of drawing network diagram, labeling Fulkerson's I-J rule, Time estimates & Critical path, PERT, Resource allocation, Uses of PERT/CPM.

References:

1. Operations Research - S D Sharma
2. Operations Research - R K Gupta & D S Hira
3. Introduction to Operation research – Frederick S Hillier ,Gerald J and Liberman.
4. Operation research: An Introduction by Hamdy A Taha.
5. Operation research: Application and algorithm by Wayne L Winston.

MOBILE COMMUNICATION

3:1:0

Course Code:CS A/B/C/D 320

Course Outcomes:

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

- CO1. Design a mobile cellular network
- CO2. Optimize a radio channel system
- CO3. Select the apt diversity scheme for a given wireless system to improve the performance.
- CO4. Perform efficient spectral allocation using multiple access techniques such as CDMA, and OFDM.
- CO5. Select the correct MAC protocol and routing algorithm for mobile ad-hoc networks.
- CO6. Optimize the mobile ad-hoc network, MAC protocols and routing algorithms as per application.

UNIT 1

Introduction, Applications, History of wireless communication, reference model, Wireless transmission, Frequencies for radio transmission, signals, Antennas, Signal propagation Multiplexing, Modulation, Spread spectrum

UNIT 2

Cellular Systems, Medium Access control, Motivation for a specialized MAC, SDMA, FDMA, TDMA, CDMA, Comparison.

UNIT 3

Telecommunications Systems, GSM, DECT, TETRA, UMS & IMT 2000
Satellite Systems, history, Applications, Basics, Classical TCP improvements, TCP over 2.5/3G wireless networks, performance enhancing proxies

UNIT 4

Support for mobility, File Systems World Wide Web, Wireless Application protocol

References:

1. Cellular and Mobile Communication- Krishna.
2. Cellular Mobile Communication – V S Bhagat
3. Cellular and Mobile Communication- V Jeyasri Arokiamary.
4. Wireless Communication and Networks – William Stallings.

5. Cellular Communication; A Comprehensive and practical guide- Nishith Tripathi and Jeffreyreed.

C++

2:0:2

Course Code:CSA250

Course Outcomes:

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

CO1 . Understand the features of C++

CO2. Understand the relative merits of C++ as an object oriented programming language

CO3. Understand how to produce object-oriented software using C++

CO4. Understand how to apply the major object-oriented concepts to implement object oriented programs in C++, encapsulation, inheritance and polymorphism

CO5 Understand advanced features of C++ specifically stream I/O, templates and operator overloading

Unit I

Quick overview of C : Expressions - Statements - Arrays and Null-Terminated Strings – Pointers - Functions – Structures, Unions, Enumerations and User-Defined Types – C Style Console I/O – File I/O -.

Unit II

An Overview of C++ - Classes and Objects – Arrays, Pointers, References, and the Dynamic Allocation Operators

Unit III

Function Overriding, Copy Constructors and Default Arguments – Operator Overloading - Inheritance – Virtual Functions and Polymorphism

Unit IV

Templates – Exception Handling - The C++ I/O System Basics

References :

JSSCACS

1. The Complete Reference C++, 4th Edition, Tata-McGraw-Hill, 2003, Herbert Schildt
2. Object Oriented Programming with C++ , M T Somashekara, D S Guru, H S Nagendraswamy and K S Manjunatha
3. C++ Premier, 5th Edition. Stanley B Lippman
4. C++ Programming language, E Balaguruswamy
5. The C++ programming language, 4th Edition, Bjarne Stroustrup

PATTERN RECOGNITION

3:0:1

Course Code: CAC230

Course Outcomes:

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

- C01. Explain and compare a variety of pattern classification, structural pattern recognition, and pattern classifier combination techniques.
- C02. Summarize, analyze, and relate research in the pattern recognition area verbally and in writing.
- C03. Apply performance evaluation methods for pattern recognition, and critique comparisons of techniques made in the research literature.
- C04. Apply pattern recognition techniques to real-world problems such as document analysis and recognition.
- C05. Implement simple pattern classifiers, classifier combinations, and structural pattern recognizers.

UNIT 1

Machine perception, pattern recognition systems, Design Cycle, Learning and adaption, models of Pattern recognition

UNIT 2

Bayesian Decision Theory

Introduction, Bayesian, Decision theory- Two category classification, classifiers-Two category case and multi category case, missing and noisy features.

UNIT 3

Nonparametric Techniques

Introduction, Density estimation, Parzen window, KN neighbor estimation, The nearest neighbor rule, Metrics and Nearest Neighbor Classification, Error analysis of nearest decision rule

UNIT 4

Introduction, Heirarchical clustering techniques, partitional clustering techniques Dimensionality reduction techniques Introduction, principle component analysis, Fisher Linear Dicriminant Analysis, Spectral clustering based dimensionality reduction

References:

1. Pattern Classification, 2nd edition, Wiley publications, R. O Duda, P.E. Hart and D G Stork,
2. Pattern Recognition and Image Analysis, Prentice Hall of India, Pvt Ltd, Earl Gose, Richard, Johnsonbaugh, Steve Jost
3. Pattern Recognition and machine Learning, Cristopher M Bishop,
4. Pattern Recognition (Blue Ant) – Willian Gibsom.
5. Pattern Recognition, 4th Edition- Sergios Theodoridis and Konstantios Koutroumbas

IMAGE PROCESSING

2:1:1

Course Code:CS A/B/C/D 330

Course Outcomes:

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

- C01. Analyze general terminology of digital image processing.
- C02. Examine various types of images, intensity transformations and spatial filtering.
- C03. Develop Fourier transform for image processing in frequency domain.
- C04. Evaluate the methodologies for image segmentation, restoration etc.
- C05. Implement image process and analysis algorithms.
- C06. Apply image processing algorithms in practical applications.

UNIT I

Introduction, digital image fundamentals

UNIT II

Image enhancement in the spatial domain, Image enhancement in the frequency domain

UNIT III

Image restoration, color image processing

UNIT IV

Wavelets and multi-resolution processing image compression

References:

1. Digital Image Processing-Rafel C.Gonzalez and Richard E Words.
2. The Image Processing hand Book- John C Cruss.
3. Fundamentals of Digital Image Processing- Anil K Jain.
4. Digital Image Processing –Jayaraman S.
5. Digital Image Processing- Sanjay M Shah Munesh Chandra Trivedi

SOFTWARE TESTING

3:0:1

Course Code:CSC440

Course Outcomes:

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

CO1.Check Various test processes and continuous quality improvement

CO2.Verify Types of errors and fault models

CO3.Check Methods of test generation from requirements

CO4.Check Behavior modeling using UML: Finite state machines (FSM)

CO5.Test generation from FSM models

CO6.Input space modeling using combinatorial designs

UNIT 1

Assessing Testing Capabilities and Competencies, Building a software Testing Environment: Building a software Testing Strategy, Establishing a Software Testing Methodology, Determining your Software Testing Techniques, Selecting and Installing Software Testing Tools.

UNIT 2

The Eleven-Step Testing Process: Eleven-Step Testing Process Overview, Step1: Access Project Management Development Estimate and status, Step2: Develop Test Plan, Step3: Requirement Phase Testing,

UNIT 3

Step4: Design Phase Testing, step 5: Requirement Phase Testing, Step6: Execute Test and Record Results,

UNIT 4

Step7: Acceptance Test Results Step8: Report Test Results, Step9: Testing Software Installing, Step10: Test Software Changes, Step11: Evaluate Test Effectiveness.

REFERENCES:

1. Effective Methods for Software Testing, William E. Perry, 2nd Edition 2003, Wiley
2. *Surviving the Top Challenges of Software Testing*, New York: Dorset House, 1997.,
Rice, Randall and Peery, William E.,
3. A practitioner's Guide to Software Test Design, By Lee Copelane.
4. The Art of Software Testing By Glenford Myers.
5. Testing Object System: Models, Patterns and Tools by Robert V Binder.

GRAPH THEORY

3:1:0

Course Code:CSB270

Course Outcomes:

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

CO1. Explain basic concepts in combinatorial graph theory

CO2. Define how graphs serve as models for many standard problems

CO3. Discuss the concept of graph, tree, Euler graph, cut set and Combinatorics.

CO4. See the applications of graphs in science, business and industry.

UNIT 1

Introduction to Graph theory: Basic terminologies—direct & undirect graphs, walks, paths & circuits, sub-graphs and complements, Graph Isomorphism, vertex degree and regular graphs,

UNIT 2

Konigsberg bridge problem & Euler graphs. Hamilton graphs & traveling salesman problem, planar graphs- definition & examples, Bipartite & Kuratowskis graphs, Euler's formula & detection of planarity, Dual of Planar graphs,

UNIT 3

Graph Coloring: Proper coloring & chromatic number of graphs, Chromatic polynomial, four color problems, Trees: Optimization & Matching: Trees; Definition & Properties, Rooted & binary rooted trees, ordered trees & trees sorting. Weighted trees & prefix codes

UNIT 4

Spanning trees, optimization, Networks, Cutset, Edge & Vertex connectivity of a graph, Max-flow Min-cut theorem and its applications, Matching theory and its applications

References:

1. Graph Theory, V.K Balakrishnan, Schaum Series, McGrawHill, 1997
2. Graph Theory, by Frank Harary, Westview Press, 1994.
3. Introduction to Graph Theory, Douglas B west.
4. Hand Book of Graph Theory, Jonathan L Gross and Jay Yellen.
5. Graph Theory with application to Engineering and Computer science, Narsingh Deo.

OOAD

2:1:1

Course Code:CSB210

Course Outcomes

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

CO1. Analyse, design, document the requirements through use case driven approach.

CO2. Identify, analyse, and model structural and behavioural concepts of the system.

CO3. Develop, and explore the conceptual model into various scenarios and applications.

CO4. Apply the concepts of architectural design for deploying the code for software
Implementation of Object Oriented concepts using C++

UNIT II

Introduction, Object orientation, OO development, OO themes, OO modeling ,History. Modeling, Abstraction, Models .Class Modeling Object & class, Link & Association concepts, Generalization & Inheritance, sample Class Model, Navigation of class models. Advanced class modeling: advanced object & class concepts, Association end N ary Association, Aggregation, Abstract Classes, Multiple Inheritance, Meta Data, Reification, Constraints, Derived Data.

UNIT III

State modeling: Events, States, Transitions & Conditions, State Diagrams, State Diagram behavior. Advanced state modeling, interaction modeling.

UNIT IV

System design: Overview of System design, Estimating performance, making a Reuse plan, Breaking a System into Subsystems, Identifying concurrency, Allocation of subsystems, Management of Data Storage.

Reference:

1. Object Oriented Analysis and Design – Blaha, Rambaugh.
2. Object Oriented Analysis and Design with the Unified Process- W Satzinger, Robert B Jackson and Stephen D Burd.
3. Object Oriented Analysis and Design with application, 3rd edition- Grady Booch, Robert A Maksimchuk, Michael W Engel
4. Object Oriented Analysis and Design with application- Grady Booch.

PROBABILITY AND STATISTICS

3:1:0

Course Code: CS A/B/C/D 340

Course Outcomes

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

- CO1. Apply probability theory to set up tree diagrams
- CO2. Apply probability theory via Bayes' Rule
- CO3. Describe the properties of discrete and continuous distribution functions
- CO4. Use method of moments and moment generating functions
- CO5. Assess the consistency, efficiency and unbiasedness of estimators
- CO6. Apply method of maximum likelihood estimation
- CO7. Apply the Central Limit Theorem
- CO8. Use statistical tests in testing hypotheses on data

UNIT 1

Introduction, basic terminology, Interpretation of probability: Axioms of probability, Some elementary theorems, Conditional probability, Mathematical Expectation

UNIT 2

Probability Distributions: Introduction, Discrete probability distributions continuous probability distributions, The expected value of a random variable Chebyshev's Theorem

UNIT 3

Sampling distributions, Populations and samples, Sampling distribution, The sampling distribution of the mean, sampling distributions of proportions, sampling distributions of mean, chi squared distribution, F distribution.

UNIT 4

Estimation and inference theory, introduction, point estimation, interval distribution, bayesian estimation, test of hypot, Introduction to ANOVA.

References:

1. Probability and Statistics: Bheeshma Rao
2. Probability and Statistics, 4th edition, Degroot, Schervish.
3. Probability and Statistics for Engineering and Science, 8th edition, Jay L Devore.
4. Probability and Statistics, Michael Akritas.

5. An Introduction to Probability and Statistics, 3rd edition, Vijay K Rohatgi and A K MD Ehsanes Saleh.

DATA MINING

2:1:1

Course Code:CSD230

Course Outcomes:

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

- CO1. Demonstrate an understanding of the importance of data mining and the principles of business intelligence
- CO2. Organize and Prepare the data needed for data mining using pre preprocessing techniques
- CO3. Perform exploratory analysis of the data to be used for mining.
- CO4. Implement the appropriate data mining methods like classification, clustering or Frequent Pattern mining on large data sets.
- CO5. Define and apply metrics to measure the performance of various data mining algorithms.

UNIT 1

Introduction, what kind of patterns can be mined? which technologies are used? which kind of applications are targeted?, major issues in Data mining.

Getting to know your data: Data objects and attribute types, basic statistical description of data, Data Visualisation, Measuring Data similarity and dissimilarity.

UNIT 2

Data Preprocessing: data cleaning, Data integration, Data Reduction, Data transformation and Data Discretization

UNIT 3

Mining frequent patterns, Associations and correlations: Basic concepts, Frequent mining methods, pattern Evaluation methods, Pattern mining in Multilevel multi dimensional space, Decision tree induction.

UNIT 4

Cluster analysis, partitioning methods, heirarchical methods, density based methods, grid based methods, evaluation of clustering.

References:

1. Data Mining: Concepts and Techniques, Third Edition, Jaiwei Han, Micheline Kamber, Jian Pei.
2. Learning Data Mining with Python, 2nd edition, Robert Layton.
3. Data Mining; The Text book, Charu C Aggarwal.
4. Data Mining, 4th edition: Practical Machine learning Tools and Techniques by Lan H Witten and Fibe Frank.
5. Introduction to Data Mining – Pang- Ning Tan and Micheal Steinbach

ARTIFICIAL INTELLIGENCE

3:1:0

Course Code:CS A/B/C/D 350

Course Outcomes

CO 1. At the end of the course students will be able to:

CO 1. Create appreciation and understanding of both the achievements of AI and the theory underlying those achievements.

CO 2. Know concepts of a Rational Intelligent Agent and the different types of Agents that can be designed to solve problems

CO 3. Review the different stages of development of the AI field from human like behavior to Rational Agents.

CO4. Impart basic proficiency in representing difficult real life problems in a state space representation so as to solve them using AI techniques like searching and game playing.

CO5. The basic issues of knowledge representation and Logic and blind and heuristic search, as well as an understanding of other topics such as minimal, resolution, etc. that play an important role in AI programs.

CO6. Introduce advanced topics of AI such as planning, Bayes networks, natural language processing and Cognitive Computing.

UNIT 1

AI problems, AI techniques, defining the problem as state space search, production systems, problem characteristics

Heuristic Search: Generate and test, hill climbing, BFS, Problem Reduction, Constraint Satisfaction, Means-End Analysis

UNIT 2

Knowledge Representation: Representations and mappings, approaches to knowledge representation

procedural v/s declarative knowledge, normal forms in predicate logic and clausal forms, non-monotonic reasoning

Declarative Representations: semantic nets, conceptual dependency, frames, scripts

UNIT 3

Game playing: minimax search procedure, adding alpha-beta cut offs

Planning: An Example Domain – the blocks world, Components, goal stack planning

UNIT 4

Expert systems: expert systems v/s conventional computers, expert system shells, explanation based learning.

Learning: Learning from observation - Inductive learning – Decision trees – Explanation based learning – Reinforcement Learning, Neural Networks, Introduction to Natural Language Processing.

References:

1. Artificial Intelligence, Third Edition, Elaine Rich, Kevin Knight, Shivashankar B Nair, Tata McGraw-Hill.
2. Introduction to Artificial Intelligence, Wolfgang Ertl.
3. Artificial Intelligence, 2nd edition, Stuart Russel, peter Norvig.
4. Artificial Intelligence, Jeorge F Luger
5. Artificial Intelligence, Saroj kaushik

.NET TECHNOLOGIES

2:0:2

Course Code:CSB280

Course Outcomes

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

- CO1. Design web applications using .NET
- CO2. Use .NET controls in web applications.
- CO3. Debug and deploy .NET web applications
- CO4. Create database driven .NET web applications and web services

Unit 1

Benefits of .NET Framework, Architecture of .NET Framework 4.0, Components of .NET Framework 4.0: CLR, CTS, Metadata and Assemblies, .NET Framework Class Library, Windows Forms, ASP .NET.

Need of C#, C# Pre-processor Directives, Creating a Simple C# Console Application, Identifiers and Keywords. Data Types, Variables and Constants: Value Types, Reference Types, Type Conversions, Boxing and Unboxing, Variables and Constants. Expression and Operators: Operator Precedence, Using the :: (Scope Resolution) Operator and Using the *is* and *as* Operators. Control Flow statements: Selection Statements, Iteration Statements and Jump Statements.

Unit 2

Arrays and Strings: One Dimensional and Multidimensional Arrays, Jagged Arrays
Classes and Objects: Creating a Class, Creating an Object, Using this Keyword, Creating an Array of Objects, Using the Nested Classes, Defining Partial Classes and Method, Returning a Value from a Method and Describing Access Modifiers. Static Classes and Static Class Members. Properties: Read-only Property, Static Property, Accessibility of accessors and Anonymous types. Indexers, Structs: Syntax of a struct and Access Modifiers for structs. Strings: Constructing Strings, Operating on Strings, Arrays of Strings, The String Class

Unit 3

Encapsulation: Encapsulation using accessors and mutators, Encapsulation using Properties.
Inheritance: Inheritance and Constructors, Sealed Classes and Sealed Methods, Extension methods. Polymorphism: Compile time Polymorphism/Overloading, Runtime Polymorphism/

Overriding. Abstraction: Abstract classes, Abstract methods. Interfaces: Syntax of Interfaces, Implementation of Interfaces and Inheritance.

Delegates: Creating and using Delegates, Multicasting with Delegates. Events: Event Sources, Event Handlers, Events and Delegates, Multiple Event Handlers. Exception Handling: The try/catch/finally statement, Checked and Unchecked Statements.

Unit 4

Introduction, Windows Forms, Life Cycle, Event Handling: A Simple Event- Driven GUI, Visual Studio Generated GUI Code, Delegates and Event- Handling Mechanism, Another Way to Create Event Handlers, Locating Event Information. Control Properties and Layout, Labels, TextBoxes and Buttons, GroupBoxes and Panels, CheckBoxes and RadioButtons, ToolTips, Mouse-Event Handling, Keyboard-Event Handling. Menus, MonthCalendar Control, ListBox Control, CheckedListBox Control, ComboBox Control, TreeView Control, ListView Control, TabControl Control ; Building an Multiple Document Interface (MDI) Application. Introduction to ADO.NET

References:

1. Programming in C# 4.0, Tata McGraw Hill, Hebert Schildt
2. C# with .net 4.0 by Andrew Troelsen
3. Programming in C# , 3rd Edition, E Balaguruswamy
4. The Complete Reference C#, Herbert Schildt.
5. The Complete Reference ASP.NET, Robert Standefer III

OBJECT ORIENTED MODELING AND DESIGN WITH UML

2:1:1

Course Code:CSA/B/C/D 360

Course Outcomes

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

- CO1. Design & Programming course is a unique course that teaches students how to use object-oriented techniques to build software.
- CO2. Gathering requirements & end with implementation.
- CO3. Analyze and design classes, their relationships to each other in order to build a model of the problem domain.
- CO4. Use common UML diagrams throughout this process, such as use-case, class, activity & other diagrams.
- CO5. Create The diagrams through a free tool.
- CO6. Capture and manage requirements.

Unit I

INTRODUCTION : Object-Oriented Analysis and Design - Iterative, Evolutionary, and Agile. - Case Studies : The NextGen POS System - INCEPTION : Inception is Not the Requirements Phase – Evolutionary Requirements - Use Cases – Other Requirements:NextGen Example

Unit II

ELABORATION ITERATION 1: Basics – Domain Models – System Sequence Diagrams – Operation Contracts – Requirements to Design-Iteratively – Logical Architecture of UML Package Diagrams

Unit III

On to Object Design – UML Interaction Diagrams – UML class Diagrams - GRASP : Designing Objects with Responsibilities – Object Design Example with GRASP:NextGen POS system – Designing for Visibility – Mapping Designs to Code

Unit IV

ELABORATION ITERATION 2: UML Tools and UML as Blueprint – Quick Analysis Update: NextGen POS – Iteration 2: More Patterns – GRASP: More Objects with Responsibilities – applying GoF Design Patterns

Reference:

1. "Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development", Third Edition, Pearson Education, 2005, Craig Larman,
2. Advanced Object Oriented Analysis and Design using UML , James J Odell.
3. Object Oriented Analysis and Design , Mike O-Docherty.
4. Object Oriented Modeling and Design with UML, 2nd edition, Michael R Blaha, James R Rumbaugh.
5. Object Oriented Systems and Analysis and Design using UML, Simon Benneth, Steve McRovv and Ray Farmer.

ANDROID APPLICATION DEVELOPMENT

2:0:2

Course Code:CSA/B/C/D 370

Course Outcomes

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

- CO1. Describe and compare different mobile application models/architectures and patterns.
- CO2. Apply mobile application models/architectures and patterns to the development of a mobile software application.
- CO3. Describe the components and structure of a mobile development framework (Google's Android Studio).
- CO4. Apply a mobile development framework to the development of a mobile application.
- CO5. Demonstrate advanced Java programming competency by developing a maintainable and efficient cloud based mobile application

Unit 1

Introduction to Android & Open Handset Alliance

Installation of Android Studio and other Development Environments like Gradle.

Understanding Android File System.

Creating First Android Application

Understanding Intent, Activity, Service, Content Providers, Broadcast Receivers.

Understanding Android Application, Manifests, Layouts, Drawables, Styles, Android Activity, View

Understanding Android XML based layout (Linear Layout, Relative Layout, Frame Layout).

Introduction to Android Life Cycle Events

initialization and Button Click Listeners.

Unit 2

Development of Simple app containing Dialog Box, Intents, Toast, Spinners, Listeners examples.

Android Listview / GridView and Adapters

Android Date Picker Dialog, Time Picker Dialog

Launching sub Activity

Building Custom ListView and Understanding Adapters in detail

Understanding SQLite database. Populating database.

Developing simple app by using SQLite database (insert, delete, update)

Unit 3

Working with web server basics

Background loading, AsyncTask , Using Threads

Developing simple app by downloading image from web and showing it in ImageView

Understanding Importance of External Libraries and demonstration of simple external library

Image lazy loading, Image loading in list view, grid view

Unit 4

Working with Google Maps

ViewPager

Introduction to fragment, add, remove, replace fragment

ViewPager

Side Navigation Drawer

Action bar/ Toolbar

ViewPager Adapter / Swipe View

References:

1. Android Programming for Beginners by John Horton.
2. Professional Android 4 application development by Reto Meir.
3. Android Book by Lan F Darwin.
4. Learning Android Building application for The Android Market by Marko Gargenta.
5. Programming Android Java programming for the new generation of Mobile Devises by Zigurd Mellieks

ADVANCED DATABASE MANAGEMENT SYSTEM

2:1:1

Course Code:CS A/B/C/D 380

Course Outcomes

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

- CO1. Evaluate and Apply Advanced Database Development Techniques.
- CO2. Evaluate Database Systems.
- CO3. Administer Database Systems.
- CO4. Design & Implement Advanced Database Systems.

Unit 1

Disk storage, Basic file Structures and hashing, indexing structures for files.
Algorithms for query optimization.

Unit 2

Physical database design and tuning, Introduction to transaction, Concurrency control techniques.
Concept for object databases, Object databases standard and design, database security.

Unit 3

Enhanced datamodels for advanced applications, distributed databases and client server architectures, Emerging database technologies and applications.

Unit 4

Definition of NoSQL, History of NoSQL and Different NoSQL products, NoSQL Basics. Exploring one among MongoDB/CouchDB/Cassandra along with Java/Ruby/Python interface : Interfacing and Interacting with NoSQL, NoSQL Storage Architecture, CRUD operations, Querying, Modifying and Managing NoSQL Datastores, Indexing and ordering datasets

References:

1. Fundamentals of Database System :5th Edition ,Navathe

2. Database Management System, Panneerselvam R.
3. Database Management System, Raghu Ramakrishnan and Johannes Gehrke.
4. Data Schemes: Models and algorithms (Advances in Database Systems), Charu C Aggarwal.
5. Multilevel secure Transaction Processing (Advances in Database system), Vijay Atluri and Sushin Jajodia.

COMPILER DESIGN

3:0:1

Course Code:CS A/B/C/D 390

Course Outcomes

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

- CO1. Construct a parse tree, or explain why no parse tree exists, given a BNF grammar and a string over the appropriate alphabet.
- CO2. Implement a lexical analyzer from a specification of a language's lexical rules.
- CO3. Compute the FIRST set for a BNF grammar.
- CO4. Compute the FOLLOW set for a BNF grammar.
- CO5. Determine FIRST intersect FIRST constraint satisfaction - determine if a BNF grammar satisfies the constraint on intersection of FIRST sets required for single-symbol-lookahead, top-down, lookahead parsing ()
- CO6. Determine FIRST intersect FOLLOW constraint satisfaction - determine if a BNF grammar satisfies the constraint on intersection of FIRST and FOLLOW sets required for single-symbol-lookahead, top-down, lookahead parsing ()

Unit-1

Introduction, Classification of grammars. Context free grammars. Deterministic finite state automata (DFA) Non-DFA.

Lexical analysis :Language processors; The structure of a Compiler; The evolution Of programming languages; The science of building a Compiler; Applications of compiler technology; Programming language basics. Lexical analysis: The Role of Lexical Analyzer; Input Buffering; Specifications of Tokens; Recognition of Tokens.

Unit-2

SyntaxAnalysis

Introduction; Context-free Grammars; Writing a Grammar. Top-down Parsing; Bottom-up Parsing.

Introduction to LR Parsing: Simple LR; More powerful LR parsers (excluding Efficient construction and compaction of parsing tables) ; Using ambiguous grammars; Parser Generators.

Unit-3

Intermediate Code Generation

Variants of syntax trees; Three-address code; Translation of expressions; Control flow; Back patching; Switch-statements; Procedure calls.

Run-Time Environments

Storage Organization; Stack allocation of space; Access to non-local data on the stack; Heap management; Introduction to garbage collection.

Unit-4

Code Generation

Issues in the design of Code Generator; The Target Language; Addresses in the target code; Basic blocks and Flow graphs; Optimization of basic blocks; A Simple Code Generator
Code optimization. Folding, redundant sub-expression evaluation. Optimization within iterative loops.

References:

1. Compilers Principles, Techniques and Tools, 2nd Edition, Addison-Wesley, 2007, Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman
- 2."The Theory and Practice of Compiler Writing". McGraw Hill, New York, 1985, Tremblay, et. al
3. Principles, Techniques and Tools of Compilers.- Allen I Holob.
4. Elements of Compiler Design.- Meduna
5. Compiler Design - K Muneeswaran.

OPEN ELECTIVES

2:0:2

OP1: COMPUTER FUNDAMENTALS

Course Code: PHC/CHC/BCC/BTC/BOC/MAC/SWC/KAC/ENC/MCC/ZOC 640

Course Outcomes:

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

- CO1. Use technology ethically, safely, securely, and legally.
- CO2. Identify and analyze computer hardware, software, and network components.
- CO3. Design basic business web pages using current HTML/CSS coding standards.
- CO4. Install, configure, and remove software and hardware.
- CO5. Use systems development, word-processing, spreadsheet, and presentation software to solve basic information systems problems.

UNIT 1

INTRODUCTION

Computer, Characteristic of Computer, History of Computer, Components of Computer
Key Factors of Computers: Hardware, Software - types of Software (Application and system), forms of software (firmware, shareware, freeware), Translator - Assembler, Compiler and Interpreters. Computer Application – Business, Scientific, Entertainment and educational.

CLASSIFICATION OF COMPUTERS

Mode of operations – Analog, Digital and hybrid Computers
Size and capabilities – Micro, Mini, Main frame and Super computer

UNIT 2

MEMORY UNITS

Primary memory - RAM, ROM, PROM, EPROM, EEPROM, Flash memory
Secondary memory – Magnetic disk (Hard disk, Zip disk, Jaz disk, Super disk)
Optical disk (CD, CD – R, CD – RW, DVD), Magneto-optical disk device

COMPUTER PERIPHERALS DEVICES AND INTERFACES

Input devices – Working principle of Keyboard and mouse, Functional capabilities of Scanner, Digital Camera, OMR, OCR, touch pad, touch screen. Output Devices – Monitor, Printer, Plotter and projector.

Processors, Classification of Processors on speed, Motherboard, Power Supply, I/O Ports and its Maintenance

PROGRAMMING LANGUAGES

Machine, Assembly language and High Level Language

UNIT 3

Introduction to Windows, Elements of Word Processing, Spread Sheet, Presentations .

Nudi/Baraha.

UNIT 4: INTERNET

Basics of Internet: www, HTTP, DNS, IP address, Email, Web browsers, Search Engines

HTML: Introduction to HTML, CSS

E-Commerce: Introduction, Types, Advantages of e-commerce, Applications, survey on popular e-commerce sites

E-governance, Introduction to Cyber Ethics

References:

1. Computer Fundamentals (6th Edition) – Rajaraman.
2. Computer's Today – Suresh K Basandra.
3. Computer Fundamentals-P K Sinha
4. Computer System Architecture (3rd Edition) PHI-2002. Chapters 3.3 & 3.4- Morris Mano,
5. Digital Principles and application (4th Edition) – Malvino Leach, Tata Mc Graw-Hill Edition
6. Computer System Architecture (3rd Edition) – Morris Mano, PHI
7. Microsoft office 365-Katherine Murray.
8. Microsoft office 2016- Nita Rutkosky, Denise Seguin, Audrey Rutkosky Roggenkamp
9. The Complete reference HTML by Herbert Schildt
10. Learn to program HTML and CSS for beginners
11. HTML black book –Steven Holzner.

