

SCHEME OF EXAMINATION
&
DETAILED SYLLABUS
(w. e. f. Academic Year 2022-2023)

for

MASTER OF COMPUTER APPLICATIONS
(MCA) DEGREE



DEPARTMENT OF COMPUTER SCIENCE
AKS UNIVERSITY, SATNA
MADHYA PRADESH, INDIA

MASTER OF COMPUTER APPLICATIONS
MCA I semester

S. NO.	PAPER CODE	SUBJECT/PAPER	Group	#L	#T	#P	TOTAL CREDIT
		THEORY					
1	44MS101	Discrete Mathematics	Math	3	1	0	4
2	44CA102	Computer Organization	Computer Science	3	1	0	4
3	44CA103	Operating System	Computer Science	3	1	0	4
4	44CA104	Computer Network	Computer Science	3	1	0	4
5	44CA105	Programming in C	Computer Science	3	1	0	4
6	44CA106	Web Technologies	Computer Science	3	1	0	4
		PRACTICAL					
1	44CA151	Programming in C –LAB	Computer Science	0	0	2	1
2	44CA152	Web Technologies-LAB	Computer Science	0	0	2	1
		TOTAL CREDIT		18	6	4	26

DISCRETE MATHEMATICS

[**PAPER CODE**]

LTP: [3|1|0]

OBJECTIVE: The basic objective of this course is to divert the mindset of students towards the mathematical foundation of computer system.

OUTCOME: Upon completion of this course the student should be able to understand the computational mathematics for developing theories of machines.

PREREQUISITE: Basic knowledge of mathematical calculus.

UNIT-I

Set Theory: Element of set, Types of set, Operation on Sets, Union, Intersection and Complement of Sets, Cartesian product, Venn diagram, Different Laws on sets. **Relation and Function:** Introduction and Types of Relation, Composition of relation, transitive composition, Symmetric-Transitive Composition, Reflexive-Transitive composition, Partial Order Relation, Equivalence Relation, Domain and Range, Onto, Into and One-One Function.

UNIT-II

Combinatory: Mathematical inductions, Strong induction and well ordering, Principle of Inclusion and Exclusion, The basics of counting, The pigeonhole principle, Permutations and combinations, Recurrence Relations, Solving Recurrence Relations, Generating Functions.

UNIT-III

Preposition: Preposition, First Order Logic, Basic Logic Operation, Logical Equivalence, Truth Table, Normal Forms, Predicates and Quantifiers, POSET, Hasse Diagram. **Lattices and Boolean algebra:** Properties of lattices, Complete Lattice, Distributive Lattice, Bounded Lattice, Lattice Homomorphism, Lattices Isomorphism, Least Upper Bound, Greatest Lower Bound.

UNIT-IV

Graph Theory: Concepts of Graph, Sub graph, Isomorphic Graph, Homomorphic Graphs, Labeled Graph, Weighted Graphs, Cyclic Graphs, Directed Graph, Complete graph, Regular graph, Bipartite graph, Connected graphs, Operations on Graphs, Matrix Representation On Graphs, Shortest Paths in weighed graphs (Dijkstra's Algo), **Tree:** Introduction of Tree, Rooted Tree, Decision trees or Sorting Tree, Spanning Tree, Binary Trees, Prim's, Algorithms and Kruskal Algorithms.

UNIT-V

Algebraic Structures: Properties, Binary operation, groupoid, semi group, monoid, Group, abelian group, Generator and evaluation of power, Subgroup, cyclic group, homeomorphism and isomorphism of group, Definition and examples of rings and field.

References:

1. Elementary Abstract Algebra 1996, B.R. Thakur Ram Prasad And Sons
2. Discrete Mathematics Aug-2006, Dr. D. C. Agrawal Shree Sai Prakeshan
3. A Textbook of Discrete Mathematics, 9th Edition, S. Chand Publishing, 2016, By Sarkar, Swapan Kumar.
4. C.L. Liu & Mahopatra, Elements of Discrete mathematics, 2nd Sub Edition 1985, Tata McGraw Hill

5. J.P. Trembley and R. P. Manohar, Discrete Mathematical Structures with Applications to Computer Science, McGraw Hill.

COMPUTER ORGANIZATION

[PAPER CODE]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the architectural foundation of computer system.*

OUTCOME: *Upon completion of this course the student should be able to understand the flow of signal and control for developing computer circuits.*

PREREQUISITE: *Basic knowledge of discrete structure and electronic devices.*

UNIT - I

STRUCTURE OF COMPUTERS: Computer Types, Functional Units, Basic Operational Concepts, Von Neumann Architecture, Bus Structures, Software, Performance, Error Detection and Correction Codes.

COMPUTER ARITHMETIC: Number Systems, Complements, Data Representation, Fixed and Floating-Point Representation, Character Representation, Addition, Subtraction, Magnitude Comparison.

UNIT - II

LOGIC GATES: Basic Logic Circuits, Logic Gates (AND, OR, NOT), Universal Gates (NAND, NOR), Ex-OR, Ex-NOR and Their Truth Tables, Laws of Boolean Algebra, De-Morgan's Theorem, Minterm, Maxterm, POS, SOP, K-Map, Simplification by Boolean expression.

UNIT - III

BASIC COMPUTER ORGANIZATION: Instruction Codes, Computer Registers, Computer Instructions and Instruction Cycle. Memory-Reference Instructions, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, CISC Vs RISC, Parallel Processing, Pipelining.

UNIT - IV

MEMORY SYSTEM: Memory Hierarchy, Semiconductor Memories, RAM (Random Access Memory), Read Only Memory (ROM), Types of ROM, Cache Memory, Performance Considerations, Virtual Memory, Paging, Secondary Storage, RAID.

UNIT - V

INPUT OUTPUT: I/O Interface, Programmed IO, Interrupt Driven IO, DMA.
MULTIPROCESSORS: Multiprocessors and Multicomputer, Characteristics of Multiprocessors, Interconnection Structures, Inter Processor Arbitration, Inter Processor Communication and Synchronization, Cache Coherence.

References:

1. Rajaraman V. and T. Radhakrishnan, Computer Organization and Architecture, Prentice Hall, 2011.
2. B. Ram Architecture and Organization, New Age International(p) limited, Publisher, third Edition.
3. Mano M. M., Digital Logic & Computer Design, 4/e, Pearson Education, 2013.
4. Patterson D.A. and J. L. Hennessey, Computer Organization and Design, 5/e, Morgan Kauffmann Publishers, 2013.
5. William Stallings, Computer Organization and Architecture: Designing for Performance, Pearson, 9/e, 2013.

OPERATING SYSTEM

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the operational approach of computer system.*

OUTCOME: *Upon completion of this course the student should be able to understand the internal operations of computer system.*

PREREQUISITE: *Basic knowledge of computer organization.*

UNIT-I

Introduction: Operating System? Operating-System Operations, Services, User and Operating-System Interface, System Calls and Types, Operating-System Design and Implementation OS Structure, Types of Systems. Case Study of Windows and Linux.

UNIT-II

Process-Process Concept, Process Control Block, Operations on Processes, Inter-Process Communication (IPC), Process-State Transition, Scheduler and its Types, CPU Scheduling – FCFS, SJF, SRTF, RR, Priority, Introduction to Process Synchronization.

UNIT-III

Threads- Threads, Multithreading Models, Threading Issues. **Deadlock:** Introduction, Deadlock Characterization, Handling of Deadlock, Deadlock Prevention and Avoidance, Banker's Algorithm, Detection and Recovery.

UNIT-IV

Memory Management-Memory Hierarchy, Contiguous and non-contiguous Memory Allocation, Swapping, Segmentation, Paging, Virtual Memory, Demand Paging, Page Replacement- FIFO, Optimal, LRU, Thrashing.

UNIT-V

Secondary Storage: Disk Structure, Disk Management, Swap-Space Management, Disk Scheduling- FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK. **File:** Concept, Operation, Attributes, Allocation Methods, Free-Space Management, Disk Access Time Evaluation.

References:

1. Operating System Concepts 8th Edition By Abraham Silberschatz, Peter B. Galvin, Greg Gagne
2. Operating Systems: Internals and Design Principles by William Stallings
3. Operating System Concepts by AviSilberschatz
4. Modern Operating Systems by Andrew S. Tanenbaum
5. Schaum's Outline of Operating Systems by J. Archer Harris

COMPUTER NETWORK

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the communication approach of computer network.*

OUTCOME: *Upon completion of this course the student should be able to understand the internetworking tools/techniques for implementing computer network worldwide.*

PREREQUISITE: *Basic knowledge of computer organization and discrete structure.*

UNIT - I

Introduction to Network: Goals and Applications of Networks, Network Structure and Architecture, OSI Reference Model, Layer Services, TCP/IP Model, Network Topology, Types, FDDI, Connecting Devices (Hub, Switch, Router, Repeater). **Physical Layer:** Framing, Transmission Media, Switching, Multiplexing, ISDN, Bit Rate, Baud Rate, Signals.

UNIT-II

Data Link Layer: MAC, Channel Allocations, CSMA, CSMA-CD, CSMA-CA, Terminal Handling, Access Control (Stop-and-Wait, ARQ, Sliding Window Protocols) Error Control Methods (Parity Checking, VRC, LRC, CRC, Hamming Code), LLC, Physical Addressing.

UNIT - III

Network Layer: Point - to- Point Networks, Routing (Static, Dynamic, Interdomain, Intra-domain Routing), Congestion Control Techniques, ARP, RARP, ICMP. **Internetworking:** IP Packet, IP Address, Subnetting, CIDR, IPv6, NAT.

UNIT - IV

Transport Layer: Design Issues, Connection Management, TCP, Handshaking, UDP, Error Control, Flow Control. Port Addressing, QoS. **Session Layer:** Design Issues, Remote Procedure Call. **Presentation Layer:** Design Issues, Data Compression Techniques, Cryptography.

UNIT-V

Application Layer: File Transfer (FTP/TFTP/HTTP), Access and Management, Electronic Mail (SMTP), DNS, Virtual Terminals. **Network Commands/Case Studies:** ping, traceroute, ipconfig, netstat, hostname, arp, telnet, NMAP, Packet Tracer etc.

References:

1. Behrouza A. Forouzen, "Data Communication and Networking", TMH
2. A.S. Tanenbaum, Computer Networks, Pearson Education
3. W. Stallings, Data and Computer Communication, Macmillan Press
4. Computer Networks by V. S. Bagad, I. A. Dhotre, Technical Publication
5. G. Shanmugarathinam, "Essential of TCP/ IP", Firewall Media

PROGRAMMING IN C

[PAPER CODE]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the logical approach of computer system.*

OUTCOME: *Upon completion of this course the student should be able to understand the logical programming for developing system software.*

PREREQUISITE: *Basic knowledge of mathematical logic.*

UNIT-I

Introduction: Programming Paradigm, C-Character Set, Identifier and Keyword, Data Types, Constants and Variables, Operators, **Decision Control Structures:** if, if-else, switch-case.

Looping Control Structures: while, do-while, for.

UNIT-II

Array: Defining, Declaring, and Initializing Array, Types of Array. **C-Functions:** Declaration and Definition, Call by Value and Call by Reference. **Storage Classes:** Automatic, External, Static, and Register, Structure and Union.

UNIT-III

Pointer: Introduction, Features, Declaring Pointer, Pointer to Array, Array of Pointers, Pointers to Function, Pointer to Structure, Pointer within Structure. **DMA:** Introduction, calloc(), malloc(), realloc().

UNIT-IV

File Handling: Introduction, Text vs Binary File. fopen(), fclose(), getc(), putc(), gets(), puts(), fprintf(), fscanf(), Random Access File, fseek(), ftell(), rewind(). **Graphics Programming:** Intergraph, Drawing Objects- Line, Rectangles Ellipse, Circle, Polygon, Text Formatting Functions.

UNIT-V

Preprocessor: Macro Substitution, File Inclusion, Conditional Compilation, Preprocessor Directive, Miscellaneous Directives, Command Line Arguments, Introduction to Objective C.

References:

1. Kernighan, Ritchie, "The C Programming Language", Prentice Hall of India.
2. Carlo Ghezzi, Mehdi Jazayeri, "Programming Language Concepts", John Wiley and Sons.
3. E. Balagurusamy, "Programming in ANSIC C", Tata McGraw Hill, 2002.
4. Yashavant Kanetkar, "Let Us C" – Seventh Edition, BPB Publications, 2007.
5. Reema Thareja, "Programming in C", Oxford University Press India, Noida.

PROGRAMMING IN C-LAB

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse digit of a number.
3. WAP to compute the sum of the first n terms of the following series $S = 1 + 1/2 + 1/3 + 1/4 + \dots$
4. WAP to compute the sum of the first n terms of the following series $S = 1 - 2 + 3 - 4 + 5 - \dots$
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given number is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
9. Write a program that swaps two numbers using pointers.
10. Write a program in which a function is passed address of two variables and then alter its contents.
11. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.
12. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
13. WAP to display Fibonacci series
 - (i) Using recursion, (ii) Using iteration
14. WAP to calculate Factorial of a number
 - (i) Using recursion, (ii) Using iteration
15. WAP for call by value and call by reference.

WEB TECHNOLOGY

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the modernism of internet technology.*

OUTCOME: *Upon completion of this course the student should be able to understand the scripting techniques for developing web based application.*

PREREQUISITE: *Basic knowledge of programming paradigm and internet.*

UNIT-I

Web Technology: Introduction, Client-Server, Scripting Languages. Basic Principles of Web Development. **HTML5:** Introduction & History, Basic Tags & Attributes, New Elements, Form Elements, Input Types, Canvas, Audio & Video, Study of Notepad++, NetBeans and Online HTML Editors.

UNIT-II

Cascading Style Sheet-Introduction, Types of CSS, Class and ID Method, DIV and SPAN Tags, Working with Border, Backgrounds, Colors, Gradients, Shadows, 2D & 3D Transforms, Transitions, Animations. **JavaScript-** JavaScript, Syntax, DOM Model, JavaScript Events (onclick, mouseout, mouse-over etc), Form Validation.

UNIT-III

XML: Fundamentals, XML Syntax, Accessing Data from XML Documents. **jQuery:** Introduction to jQuery, Structure of jQuery, Including .js file to HTML, Selectors, Events, jQuery Effects (Hide/Show, Fade, Animate).

UNIT-IV

PHP: Introduction, Syntax, Comments, Variables and Constants, Embedding PHP in HTML, Form validation, User Defined Functions, Strings functions, Creating Array and Accessing Array Elements, Control Statements, Loops.

UNIT-V

Object Oriented Programming with PHP- Creating Classes and Objects, Accessing Methods. **CRUD:** Database Creation, Inserting, Deleting and Updating Records, Session and Cookies. **PHP with MS-Access/MySQL:** Creating Connections, Selecting Database, Perform Database Operations.

References:

1. Beginning PHP5, Apache, and MySQL Web Development 2005 Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass Wrox Publication
2. Beginning HTML, XHTML, CSS, and JavaScript 2010 Jon Duckett Wiley Publishing, 2010.
3. Web Technologies, Black Book, Dream Tech Press 2010 Kogent Learning Solutions Inc Dream Tech Press.
4. Web Technologies: TCP/IP, Web/ Java Programming, and Cloud Computing, 3e Kindle Edition.
5. HTML and CSS: Design and Build Websites Hardcover – 30 May 2014 by Jon Duckett.

WEB TECHNOLOGIES-LAB

1. Create a web page by making use of the following tags: HEAD, BODY, BGCOLOR.
2. Write an HTML page that contains a selection box with a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).
3. Create an External Style Sheet to defines the style for the following tag: H1, H2, Body, P.
4. Create an Internal Style Sheet that defines a style for Positioning elements & editing the background (color/image).
5. Create a Web page Layout with Tables and all its attributes.
6. Write a JavaScript program to design a simple calculator.
7. Write a JavaScript program to find the factorial of given number by using function.
8. Write a JavaScript program to form validation in HTML.
9. Write a JavaScript program to demonstrate the use of message box, and loops.
10. Demonstrate JavaScript events.
11. Create a web form using PHP for Student Registration and Login.
12. Write a jQuery script to perform hide /show effect on login and logout button?
13. Create a XML document with following details: rollno, sname, contact, email, address.
14. Write a PHP script to perform CRUD operations.
15. Write a JavaScript Program to demonstrate Functions (predefined / user defined)

MASTER OF COMPUTER APPLICATIONS
[MCA]
SECOND SEMESTER

PAPER CODE	PAPER NAME	#L	#T	#P	TOTAL CREDIT
	Theory				
44CA201	Object-Oriented Programming using C++	3	1	0	4
44CA202	Database Management System	3	1	0	4
44CA203	Data Structure	3	1	0	4
44CA204	.NET Technologies	3	1	0	4
44CA205	Cloud Computing	3	1	0	4
44CA206	Elective- I	3	1	0	4
	Practical				
44CA251	Programming in C++ –LAB	0	0	2	1
44CA252	.NET Technologies- LAB	0	0	2	1
44CA253	Data Structure & DBMS- LAB	0	0	2	1
	TOTAL CREDIT	18	6	6	27

Elective I-

- 44CA206-A. Linux Administration
- 44CA206-B. AI & Machine Learning
- 44CA206-C. Artificial Intelligence for Real World Application (AI) TCS iON.
- 44CA206-D. SWAYAM Subject

OBJECT ORIENTED PROGRAMMING USING C++

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the bottom-up approach of real world problems.*

OUTCOME: *Upon completion of this course the student should be able to understand the principles of object-class for developing oops based application.*

PREREQUISITE: *Basic knowledge of programming paradigm.*

UNIT – I

OOP: Programming Paradigms, Basic Concepts, Benefits of OOP, Applications of OOPs, History of C++, Structure of C++, Basic Data Types, Type Casting, Input and Output Statements, Classes and Objects, Defining Class and Member Function Specification, Scope Resolution Operator, Access Qualifiers, Instance Creation, Namespace.

UNIT-II

Functions: Function Prototyping, Call by Reference, Return by Reference, inline Functions, Default Arguments, Function Overloading, Function Overriding, Friend Class and Function, **Pointers:** Array of Objects, Pointers to Objects, this Pointer, Dynamic Allocation Operators, Dynamic Objects.

UNIT – III

Constructors: Constructors, Different Types of Constructors, Overloading Constructors, Destructors, Container Classes, Static Class Members, Static Objects. **Operator Overloading:** Overloading Unary and Binary Operator, Stream Operator Overloading, Data Conversion.

UNIT – IV

Inheritance: Defining Derived Classes, Types of Inheritance, Abstract Classes. **Polymorphism:** Introduction, Virtual Function, Virtual Base Class, Pure Virtual Function. **Templates:** Function Template, Class Template.

UNIT –V

Manipulators: Manipulators, User Defined Manipulators, IO Functions. **Exception Handling:** Handling User-Defined Exception, try/catch/throw. **Streams:** Stream Classes, File Streams, File Open and Close.

Text Books -

1. Complete Reference of C++ By Herbert Schilder
2. Object Oriented Programming with C++ By E. Balaguruswamy
3. K R Venugopal, Rajkumar Buyya, "Mastering C++", Tata Mc Graw Hill, 2013.
4. Object Oriented Programming in C++, Robert Lafore, 2000 Edition, GALGOTIA
5. Programming with C Plus Plus by D. Ravichandra

OBJECT ORIENTED PROGRAMMING USING C++-LAB

1. Write a program in C++ to exchange the content of two variables using call by reference
2. Write a program in C++ to demonstrate the Constructor Overloading.
3. Write a program in C++ to create the class shape, and overload the function to return the perimeters of the different shapes.
4. Write a program in C++ demonstrating the public, protected and private parameters.
5. Write a program in C++ to search the 2nd largest element in an array.
6. Write a program in C++ to demonstrate constructor with default argument.
7. Write a program in C++ to demonstrate multiple inheritance.
8. Write a program in C++ to demonstrate constructor call in the derived class.
9. Write a program in C++ to copy the content of file into another.
10. Write a program in C++ to append the content of the file.
11. Write a program in C++ to demonstrate virtual function.
12. Write a program in C++ to demonstrate friend function.
13. Write a program to implement an Account Class with member functions to Compute Interest, Show Balance, Withdraw and Deposit amount from the Account.
14. Write a C++ program to implement a student class having roll no., name, rank, addresses as data members.
15. Write a program to add two complex number using operator overloading.

DATABASE MANAGEMENT SYSTEM

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the management approach of database.*

OUTCOME: *Upon completion of this course the student should be able to understand the information organization for creating and managing data and record.*

PREREQUISITE: *Basic knowledge of computer system and storage.*

UNIT-I

Purpose and Advantages of DBMS, View of Data, DBMS Architecture and Data Independence, Database Languages. Classification of DBMS, Schema and Sub Schema. Database Administrator and Users, Data Dictionary, Data Modeling using ER Model, Entities, ER Diagrams, Specialization and Generalization.

UNIT-II

Attributes and Relationships, Different Types of Attributes, Domains, Database Keys - Candidate, Primary, Alternate and Foreign Keys. Codd's Rule, Relational Algebra, Tuple Relational Calculus, Domain Relational Calculus, Aggregate Functions, Null Values, Join Relations.

UNIT-III

Introduction to Normalization, Lossless Decomposition, Functional Dependency, 1st, 2nd, 3rd BCNF, 4NF, 5NF. Introduction of Relational Database Design, Trivial and Non-Trivial Dependencies, Closure Set of Dependencies. **Basic SQL:** DDL, DML and DCL Commands, Specifying Constraints in SQL, Select Statement, Additional Features of SQL, PL/SQL, Cursor, Trigger, View.

UNIT-IV

Transaction Management- Basic Concepts, ACID Properties, Transaction States, Implementation of Atomicity and Durability. Basic Idea of Serializability. Concurrency Control- Lock Based Protocols, Time Stamp Based Protocols, Validation Based Protocols.

UNIT-V

Advances in DBMS- Meaning of Deductive Databases, Technology of Multimedia Databases, Overview of Digital Libraries, Mobile Databases, Distributed and Parallel DBMS. B+ Tree in Database.

References:

1. Database Management Systems: Raghu Ramakrishnan
1. Ivan Byross, Programming in PL/SQL,
2. Elmasri & Navathe, Fundamentals of Database Systems, 7th Edition, Pearson education
3. Korth, Sudarshan, Database system concepts, McGraw hill
4. Database Management Systems – Alexis Leon, Mathews Leon – Leon, Vikas Publications
5. Oracle 9i The Complete Reference – Kevin Loney, George Koch - Oracle Press

DATABASE MANAGEMENT SYSTEM-LAB

1. Explain SELECT statement in detail.
2. Write a program to find factorial given number.
3. Write a program to find largest number among three numbers.
4. What is cursor? How will you implement implicit cursor?
5. What is function? Write five built in function with example.

DATA STRUCTURE

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the mechanisms of data flow.*

OUTCOME: *Upon completion of this course the student should be able to understand the techniques for implementing data flow models.*

PREREQUISITE: *Basic knowledge of logical programming and computer organization.*

UNIT-I

Introduction to Data structures: Definition, Classification and Operations on data structures, DMA, Asymptotic Notation, Algorithm Complexity. Big O Notation. **Linked List:** Linked List, Types of Linked List, Singly linked List, Doubly Linked List, Circular Linked list, Circular Doubly Linked List. Application of Linked List-Polynomial Representation and Addition.

UNIT-II

Stack: Stack, Array and Linked List Representation of Stack, Operations on Stack, PUSH and POP. Applications of Stack, Conversion from Infix to Postfix and Prefix. Evaluation of Prefix and Postfix Expression using Stack, Recursion. **Queue:** Array and Linked List Representation of Queue. Types of Queue, Various Operations on Queue. Applications of Queue.

UNIT-III

Graphs: Introduction, Graph Representation-Adjacency Matrix, Adjacency List, Adjacency Multi-list, Graph Traversal, Minimum Spanning Tree, Shortest Path Algorithm, Kruskal and Prim's Algorithm.

UNIT-IV

Trees: Introduction, binary tree, Binary Tree Representation, Complete Binary Tree, Extended Binary Tree, Array and Linked List Representation of Binary Tree, Traversing Binary Trees, Threaded Binary Tree, B-tree, 2-3 tree, AVL Tree, Insertion and Deletion in Binary Search Tree, Heap Introduction, Max Heap and Min Heap.

UNIT-V

Sorting: Introduction, Types of Sorting -Selection, Bubble, Insertion, Merge, Heap, Quick Sort. **Searching:** Introduction, Linear and Binary Search. **Hashing:** Basics, Methods, Collision Resolution, Chaining, Linear Probing, Rehashing. **Advances in Data Structure:** Greedy Method, Knapsack Problem, Dynamic Programming, Travelling Salesman Problems.

References:

1. Data structure – A Pseudocode Approach with C – Richard F Gilberg Behrouz A. Forouzan, Thomson.
2. G.S. Baluja, Data Structure and Algorithms
3. Advanced Data structures, Peter Bras
4. Schaum's Outlines Data structure Seymour Lipschutz Tata McGraw Hill 2nd Edition.
5. "An Introduction to Structure with application" Jean – Paul Trembly and Paul Sorenson.

.NET TECHNOLOGIES

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards graphical user interface of computer system.*

OUTCOME: *Upon completion of this course the student should be able to understand the in-built controls for developing visual and interactive application.*

PREREQUISITE: *Basic knowledge of GUI-Application.*

UNIT-I

Introduction to .Net - Features of .Net, .Net Framework Architecture and Components, .Net Code Execution Process, MSIL, Assemblies and Meta Data, Garbage Collection, Name Spaces Different types of .Net Applications, IDE of .Net, .Net Application Types, Creation of first .net Application.

UNIT-II

Introduction to VB.Net: VB.net windows form introduction, Windows form Events and properties, Windows control: TextBox, Label, Button etc., Creating classes in VB.NET, Events in VB.net, Modal and non-Modal forms, Dialog controls, MDI Form.

UNIT-III

C#.Net: Overview of C#, C#.Net Windows and Console Application, Processing Console Input and Output. **C# and OOPs:** Class and Object, Scope and Visibility, Constructors, Overloading, Inheritance, Exception Handling, File I/O, Multithreading.

UNIT-IV

Web Based Software Development: Introduction to Web Servers, ASP.NET Web Form Controls: TextBox, Label, Button etc., **Validation Controls:** Required Field Validator, Range validator, Regular Expression Validator, Compare Validator. Session and State Management.

UNIT-V

Introduction to XML: Accessing Data from XML Document, Introduction to Web Services, Creation of Web Service. **ADO.Net:** Architecture of ADO.Net-Connected and Disconnected, .Net Data Providers, Data Access: Using Command- Data Reader and Data Adapter-Data Set, Binding Data in Data Grid View, Operation on Data Grid Control.

References:

1. ASP.Net 3.5 Black Book (Covers C# and VB 2008 Codes) – Dream Tech Publication
2. The Complete Reference ASP.Net By Mathew Macdonald – TMH
3. Kothari Nikhil and Datye Vandana, Developing ASP .NET Server Controls and Components, Tata McGraw Hill, 2003.
4. Esposito Dino, Applied XML Programming for Microsoft .NET, Tata McGraw Hill, 2003.
5. C# Using .Net Framework by Lalit Arora, Anjali Arora

.NET TECHNOLOGIES-LAB

1. Write a VB.Net program for calculator.
2. Write a VB.Net program for implementation of class.
3. Write a program to implement MDI
4. Implementation of dialog boxes.
5. Implementation of Modal and non-Modal windows form.
6. Write C# code to implement inheritance.
7. Write code to implement Exception Handling
8. Write C# program to implement operator overloading.
9. Write C# code to implement multithreading
10. Write C# code to write and read data to and from file
11. Create a web page with use of different validation controls.
12. Write code for ADO implementation.
13. Write a program to access data from XML file.
14. Write code to create and use of web service
15. Write code to use different commands in GridView control

CLOUD COMPUTING

[PAPER CODE]

LTP: [3|1|0]

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the sharing of computing and storage devices.*

OUTCOME: *Upon completion of this course the student should be able to understand the suitability and advantages for using shared server and machine.*

PREREQUISITE: *Basic knowledge of server and networking.*

UNIT I

Cloud Computing Fundamentals – Introduction of Cloud Computing, History of Cloud Computing, Key Characteristics, Private, Public and Hybrid Cloud. Cloud types; IaaS, PaaS, SaaS. Benefits and Challenges, Cloud Computing vs. Cluster Computing vs. Grid Computing; Applications of Cloud Computing, Deploying a web service in Cloud platform.

UNIT II

Virtualization for Cloud– Need for Virtualization, Pros and Cons, Types of Virtualization, Virtual Machine Monitor and its Types, Virtual Machine Properties, Interpretation and Binary Translation, Hypervisors – Introduction, Hypervisor in Virtualization, Types of Hypervisor, Xen, KVM, VMWare, Virtual Box, Hyper-V.

UNIT III

Cloud File System – GFS, HDFS, BigTable, HBase, Dynamo, Comparison among File Systems; MapReduce: Introduction, MapReduce Model, Relational Operations using MapReduce, Applications of MapReduce.

UNIT IV

Cloud Security and Standards - Cloud Security Challenges, Software as a Service Security; Common Standards: The Open Cloud Consortium, The Distributed Management Task Force, Standards for Application Developers, Standards for Messaging, Standards for Security, End User Access to Cloud Computing, Mobile Cloud Computing.

UNIT V

Cloud Vendors: Amazon AWS Cloud, Google App Engine, Microsoft Azure, Salesforce. Service Management: Service Level Agreements (SLAs), Billing & Accounting, Load Balancing in Cloud; Cloud Web Technologies: Introduction, SOAP, REST, SOAP vs REST, AJAX.

References:

1. Raj Kumar Buyya, Vecchiola, and S. Thamarai Selvi, “Mastering Cloud Computing”, McGraw Hill Education Pvt. Ltd.
2. Gautam Shroff, “Enterprise Cloud Computing”, Cambridge University Press.
3. “Cloud Computing Black Book” Dreamtech Press.
4. “Cloud Computing and Virtualization Technologies in Libraries” IGI Global.
5. “Resource Management and Efficiency in Cloud Computing” IGI Global.

LINUX ADMINISTRATION

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the communication foundation of computer network.*

OUTCOME: *Upon completion of this course the student should be able to understand the administrative policies for server and network management.*

PREREQUISITE: *Basic knowledge of storage and networking.*

UNIT-I

Introduction: History, What is LINUX, Basic Architecture, Different Flavour, CUI And GUI LINUX Vs Windows File System and Blocks, Installation of Linux. **Essential Tools:** Log in and Switch Users, Create and Edit Text Files, Delete, Copy, and Move Files and Directories, Create Hard and Soft Links, Archive, File Compression and Decompression using tar, star, gzip, and bzip2.

UNIT-II

Services: Boot, Reboot, and Shut Down, Start and Stop Virtual Machines, IP Configuration, Start, Stop, and Check the Status of Network Services. **Configure Local Storage:** List, Create, Delete Partitions, Logical Volumes, Swap. **File System Configuration:** Create, Mount, Unmount, vFAT, ext4, xfs File Systems, Extend Existing Logical Volumes, Create and Manage Access Control Lists.

UNIT-III

Shell Programming: Shell, Types of Shell, Shell Variable, Keywords, Environment Variable, Shell Script, Parameter Passing, Positional Parameter & Shifting, For Loop, While Loop, Until Loop, If Statement, Case Statement. **User and Group Management:** Create, Delete, Modify Local User Accounts, Create, Delete, Modify Local Groups and Group Memberships. **Text Manipulation:** grep, egrep, sed, cut, paste, sort, split, **User to User Communication:** write, mail, msg, wall.

UNIT-IV

Security Management: Configure Firewall, firewall-config, firewall-cmd, iptables, Set Enforcing and Permissive Modes for SELinux. **Process:** Process, Types, Process Command: ps, kill, nice. **Scheduling Commands:** at, crontab, sleep, wait, Back Ground Jobs. **AWK:** Pattern Scanning, BEGIN and END Pattern, awk Arithmetic, Variables, Operations and Function.

UNIT-V

IPC & Socket Programming, Webhosting, FTP, TELNET, Traceroute, DNS, Linux and Cloud Introduction to Cluster and Site, Docker Container, OpenShift, Kubernetes, Ansible. Azure Introduction to Scripting Language Python.

References:

1. MJ Bach “Design of Unix OS”
2. Meeta Gandhi, Tilak Shetty and Rajiv Shah – The ‘C’ Odyssey Unix –The open Boundless C ,1st Edition, BPB Publications 1992.
3. Y Kanetkar “Unix shell programming”, BPB Pub
4. Rachel Morgan. Henry McGilton “Introducing UNIX System V”
5. RHCSA/RHCE Red Hat Linux Certification Study Guide, Seventh Edition by Michael Jang.

LINUX ADMINISTRATION-LAB

1. Study of requirement and Linux installation.
2. Working with File related commands.
3. Working with Directory related commands.
4. Mounting of file system and command.
5. Editing a file with vi.
6. Working with user and group
7. Implementation of Job scheduling.
8. Add two numbers using –
 - A. Shell Script
 - B. Python
9. Shell script to print the greatest of three numbers.
10. Program to add two numbers using parameter passing.
11. Program to implement AWK.
12. Program to implement IPC.
13. Creation of different partition.
14. Create and host a webpage.
15. Implementation of OpenShift technology

DATA SCIENCE AND VISUALIZATION

[PAPER CODE]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the analytical approach of data visualization.*

OUTCOME: *Upon completion of this course the student should be able to understand the statistical techniques for data visualization.*

PREREQUISITE: *Basic knowledge of statistics.*

UNIT I

Introduction to Data Science, Exploratory Data Analysis, Linear Regression and Regularization, Model Selection and Evaluation, Classification, kNN, Decision Trees, SVM, Ensemble Methods, Random Forests, Intro to probability, Naïve Bayes and Logistic Regression, Feature Engineering and Selection.

UNIT-II

Clustering, k-means, Hierarchical Clustering, Dimensionality Reduction, PCA and SVD, Text Mining and Information Retrieval, Recommender Systems, Big Data Storage and Retrieval, noSQL, GraphDB, Big Data Distributed Computing, MapReduce, Spark RDD.

UNIT III

Introduction of Visual Perception, Visual Representation of Data, Gestalt Principles, Information Overloads, Creating Visual Representations, Visualization Reference Model, Visual Mapping, Visual Analytics, Design of Visualization Applications.

UNIT IV

Classification of Visualization Systems, Interaction and Visualization Techniques Misleading, Visualization of One, Two and Multi-Dimensional Data, Text and Text Documents.

UNIT V

Visualization of Groups, Trees, Graphs, Clusters, Networks, Software, Metaphorical Visualization, Visualization of Volumetric Data, Vector Fields, Processes and Simulations, Visualization of Maps, Geographic Information, GIS Systems, Collaborative Visualizations, Evaluating Visualizations.

References:

1. Ward, Grinstein Keim, Interactive Data Visualization: Foundations, Techniques, and Applications. Natick: A K Peters, Ltd.
2. Jake VanderPlas, "Python Data Science Handbook", O'REILLY.
3. Allen B. Downey, "Think Python", O'REILLY.
4. Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, "Introduction to Statistical Learning", Springer.
5. Julie Steele, Noah Iliinsky, "Beautiful Visualization, Looking at Data Through the Eyes of Experts", O'REILLY.

MASTER OF COMPUTER APPLICATIONS
[MCA]
THIRD SEMESTER

PAPER CODE	PAPER NAME	#L	#T	#P	TOTAL CREDIT
	Theory				
44CA321	Programming in Java	3	1	0	4
44CA322	Elective- II	3	1	0	4
44CA323	Elective- III	3	1	0	4
44CA324	TOC & Compiler Design	3	1	0	4
44CA325	Software Engineering	3	1	0	4
44CA326	Cryptography & Network Security	3	1	0	4
	Practical				
44CA371	Programming in Java-LAB	0	0	2	1
44CA372 A/B	Elective- II-LAB	0	0	2	1
44CA373	Minor Project- LAB	0	0	2	1
	TOTAL CREDIT	18	6	6	27

Elective -II

- 44CA322-A. Python & R Programming
- 44CA322-B. Android Programming

Elective-III

- 44CA322-A Applied Cloud Computing (CC) – TCS iON
- 44CA322-B IoT and its Applications (IOT)- TCS iON
- 44CA322-C SWAYAM Subject
- 44CA322-D Data Science and Visualization

JAVA PROGRAMMING

[PAPER CODE]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the computational problems and solutions of real world.*

OUTCOME: *Upon completion of this course the student should be able to understand the IDE/platform for developing console and web-based applications.*

PREREQUISITE: *Basic knowledge of OOPs and DBMS.*

UNIT-I

Introduction to Java: Introduction, Features of Object-Oriented Programming (OOP), Java Virtual Machine, Byte Code, Data Types, Variable, Arrays, Expressions, Operators, Control Statements, Iteration Statements.

UNIT-II

Objects and Classes, Access Control, Constructor, Constructor Overloading, Finalize, Method Overriding, Inheritance, Abstract Class, Package, Interfaces.

UNIT-III

Exception Handling: try, catch, throw, throws, finally; Multithreading, Thread Life Cycle, Advantages and Issues, Thread Synchronization; Input Streams, Output Streams, Object Serialization, Deserialization, String Handling.

UNIT-IV

Introduction to AWT: Programming Layout and Component Managers, Event Handling, Applet Class, Applet Life-Cycle Passing Parameters Embedding in HTML. Swing Components JApplet, JButton, JFrame, etc. Sample Swing Programs. **[12 Hours]**

UNIT-V

Database Connectivity: JDBC Architecture, Establishing Connectivity and Working with Connection Interface, Working with Statements, Creating and Executing SQL statements, Working with Result Set, Handling Database Queries. **[12 Hours]**

References:

1. Programming with Java, A Primer E. Balguruswami
2. Java- The Complete Reference Patric Naughton, Herbert Schildt
3. Java Programming John P. Flynt Thomson 2nd.
4. Java Programming Language Ken Arnold Pearson.
5. Core Java: An Integrated Approach, Rao R. Nageswara

JAVA PROGRAMMING-LAB

1. Program to Create, Insert, Delete and Display Operations on Single Linked List.
2. Program to Create, Insert, Delete and Display Operations on Double Linked List.
3. Program to Create, Insert, Delete and Display Operations on Circular Single Linked List.
4. Program to Split a Single Link List.
5. Program to Reverse a Single Linked List.
6. Program to Implement PUSH and POP Operations on Stack using Array Method.
7. Program to Implement PUSH and POP Operations on Stack using Linked List Method.
8. Program to Implement Insert and Delete Operations on Queue using Array Method.
9. Program to Implement Insert and Delete Operations on Queue using Linked List Method.
10. Program to Implement Insert and Delete Operations on Priority Queue.
11. Program to Implement Insert and Delete Operations on Double Ended Queue.
12. Program to Evaluate Postfix Expression by using Stack.
13. Program to Construct Binary Search Tree and Implement Tree Traversing Techniques.
14. Program to Implement Insertion Sort.
15. Program to Implement Selection Sort.

PYTHON AND R PROGRAMMING

[PAPER CODE]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the analytical techniques of data science and machine learning.*

OUTCOME: *Upon completion of this course the student should be able to understand the methods/algorithms for developing data analytics tools.*

PREREQUISITE: *Basic knowledge of using packages and libraries.*

UNIT – I

Introduction: History, Features, Working with Python, Basic Syntax, Variable and Data Types, Operator. Conditional Statements, Looping, Control Statements, String Manipulation.

UNIT – II

Lists: Introduction, Accessing list, Operations, Working with lists, Function and Methods

Tuple: Introduction, Accessing Tuples, Operations, Working, Functions and Methods.

Dictionaries: Introduction, Accessing values in dictionaries, working with Dictionaries, Properties, Functions.

UNIT-III

Input-Output: Opening and Closing file, Reading and Writing files, Creating and Accessing Modules. **Exception Handling:** Exception, Exception Handling, User Defined Exceptions.

OOPs concept: Class and object, Attributes, Inheritance, Overloading, Overriding, Data hiding.

UNIT-IV

Libraries for Data Analytics, Numpy: Creating NumPy arrays, Indexing and slicing in NumPy. **Scipy:** Create function, modules of SciPy. **Pandas:** Series and Data Frames, Using multilevel series. **Matplotlib:** Basic plots in Matplotlib, Working with Images.

UNIT-V

R Programming: Introduction, Advantages and Disadvantages, Basic Syntax, Data Structures, Variables and Operators in R, R if Statement, for loop, while loop, R repeat loop, R Functions, Data Visualization and Statistics.

References:

1. Programming in Python, Mark Summerfield, 2nd edition, Addison – Wesley publication
2. R for Data Science", Hadley Wickham, O'Reilly
3. Programming Python, 4th Edition - O'Reilly Media
4. Advanced R, Hadley Wickham
5. Data Visualization: A practical introduction, by Kieran Healy

PYTHON AND R PROGRAMMING-LAB

1. Write a Python program to retrieve string in reverse order.
2. Write a Python Program to find area of a circle.
3. Write a Python program to check whether a number is Prime or not.
4. Write a Python program to find largest element in an array.
5. Write a Python program to interchange first and last elements in a list.
6. Write a Python program to implement Matplotlib library.
7. Write a Python program to check if a string is palindrome or not.
8. Write a Python program to reverse words in a given String in Python.
9. Write a Python program to create grade calculator in Python.
10. Write a Python program for creating a Numpy array.
11. Python program to check if a string contains all unique characters.
12. Write a Python Python program to read and write data from a file.
13. Write an R Program to Visualize Pie Charts.
14. Write an R program to create a list containing strings, numbers, vectors and logical values.
15. Write an R program to find all elements of a given list that are not in another given list.

ANDROID PROGRAMMING

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the android environment of smart phones.*

OUTCOME: *Upon completion of this course the student should be able to understand the programming techniques/tools for developing android application.*

PREREQUISITE: *Basic knowledge of Java-Programming and Applications of Smart Phones (GUI).*

UNIT-I

Introduction to Android: History and Evolution, Introduction to IDEs (Android Studio, Eclipse), Installation of Android Studio, Creation of AVD (Emulator), Dalvik Virtual Machine & .apk file extension. **Concepts of JAVA:** OOPs, Exception Handling, Packages & Interfaces, Multi-Threading (Thread Class; Runnable Interface).

UNIT-II

Basic Building Blocks-Activities, Services, Broadcast Receivers & Content Providers, Androidmanifest.xml, Uses-Permission, Activity Registration, SDK. **Resources & Layouts-**Drawable Resources, Activities and Activity Lifecycle, First Sample Application, Layouts (Linear,Relative, Grid, Frame, Constraint, Drawer).

UNIT-III

UI Design: Form Widgets, Text Fields, Button Controls, Toggle Buttons, AlertDialog, CheckBox, Spinner, TextView, EditText, RatingBar, DatePicker, TimePicker. **Android Menu:** Study of Option Menu, ContextMenu, Popup Menu.

UNIT-IV

Content Providers: Introduction to Databases in Android Programming, Shared Preferences Database, Insert, Delete and Update Data using Shared Preferences, SQLite Database, SQLiteOpenHelper, CRUD using SQLite, Introduction to Firebase Database.

UNIT-V

Network Connectivity: Restful Webservice, Get and Post Methods, Creation of API, Domains and Servers, Json, Connect Server with Application, Retrieve Data from Server, Upload Image on Server, CRUD Operations on Server.

References:

1. Android Programming: Pushing the Limits, Erik Hellman, Wiley publications.
2. Android application development for java programmers. By James C. Sheusi. Publisher: Cengage Learning, 2013.
3. Head First Android Development by Dawn Griffiths and David Griffiths, O'RELLY.
4. Beginning Android Application Development by Wei-Meng Lee.
5. The complete Reference JAVA 7th edition by Herbert Schildt.

ANDROID PROGRAMMING-LAB

1. Create —Hello World application. That will display —Hello World in the middle of the screen in the emulator. Also display —Hello World in the middle of the screen in the Android Phone.
2. Create a simple application to switch the activity.
3. Create a simple calculator.
4. Create a menu with 5 options and selected option should appear in text box.
5. Create simple application to call a fragment.
6. Create an application to run background services
7. Create a list of all courses in your college and on selecting a particular course teacher-in-charge of that course should appear at the bottom of the screen.
8. Create an application with three option buttons, on selecting a button color of the screen will change.
9. Create and Login application. On successful login, pop up the message.
10. Create an application to insert and delete by using shared preferences.
11. Create an application to Create, Insert, update, Delete and retrieve operation on the SQLite database.
12. Create an application for camera with flash and other options.
13. Create an Application to retrieve data from server.
14. Create an application for music player.
15. Create an application for video player.

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the artificial machine intelligence of computer system.*

OUTCOME: *Upon completion of this course the student should be able to understand the techniques for developing artificially intelligent agent.*

PREREQUISITE: *Basic knowledge of algorithm development and implementation.*

UNIT-I

Artificial Intelligence: Definition, Branches, Applications. **Intelligent Agent:** Rational Agent, Reflex Agent, Goal Based Agent, Utility Based Agent. **Problem Solving:** General Problem Solving, Constraint Satisfaction Problem, Exhaustive-Search and Heuristic Search, A* and AO* Algorithms, Mini-Max and Alpha-Beta Pruning.

UNIT-II

Reasoning System: Symbolic Reasoning, Statistical Reasoning, Bayes Theorem, Dempster-Shafer Theorem. **Knowledge Representation:** Knowledge Engineering, Propositional Logic and Predicate Logic, Production Rule, Semantic Network, Frame System, Scripts.

UNIT-III

Learning System: Supervised Learning, Unsupervised Learning, Reinforcement Learning, Genetic Learning. **Expert System:** Definition, Architecture, Components, Applications, Knowledge Base, Inference Engine, Working Memory.

UNIT-IV

Machine-Learning: Definition, Types of Learning. Linear Regression, Decision Trees, Overfitting. Probability and Bayes Learning. Logistic Regression, Support Vector Machine, Kernel Function and Kernel SVM. PAC Learning Model, Ensemble Learning, Recommendation. Clustering: k-means, Adaptive Hierarchical Clustering, Gaussian Mixture Model.

UNIT-V

Deep-Learning: Introduction to Deep Learning, Neural Network Basics, Shallow Neural Network, Deep Neural Network, Practical Aspects of Deep Learning, Optimization Algorithms, Hyperparameter Tuning, Batch Normalization, **TensorFlow-Keras:** Foundations of Convolutional Neural Network, Deep Convolutional Models, Recurrent Neural Networks.

References:

1. Artificial Intelligence: A Modern Approach by Stuart Russel and Peter Norvig.
2. Instruction to Artificial Intelligence by Philip C Jackson
3. Artificial Intelligence, Elaine Rich, Kevin Knight, Mcgraw Hill
4. Introduction to Machine Learning with Python by O'relly
5. Machine Learning by Tom M. Michel

THEORY OF COMPUTATION AND COMPILER DESIGN

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the computational phenomenon of machines.*

OUTCOME: *Upon completion of this course the student should be able to understand the techniques for developing computational models.*

PREREQUISITE: *Basic knowledge of discrete structure.*

UNIT-I

Introduction to Computational Science: Definition of Alphabet, Word/String, Language, Grammar, Automata and its Types, Formal Languages and its Types, Grammar and its Types, Chomsky Hierarchy for Formal Languages and Automata, Closure Properties. **Introduction to Compiler:** Compiler, Cousins of Compiler, Phases of Compiler, Symbol Table.

UNIT-II

Finite State Automata: Representation of Finite State Automata, String/Language Acceptability in Finite State Automata, NFA to DFA Conversion, Minimization of DFA. **Regular Expression:** Rules, Identities, Arden's Theorem, Simplification of Regular Expression using Identities, DFA to Regular Expression Transformation, Regular Expression to DFA, Pumping Lemma for Regular Language.

UNIT-III

Push-Down Automata: Representation of Push-Down Automata, String/Language Acceptability by Push-Down Automata. **Context Free Language and Grammar:** Parse/Derivation Tree, Right-Most Derivation, Left-Most Derivation, Ambiguities in Context Free Grammar, Pumping Lemma for CFL, Chomsky Normal Form, Greibach Normal Form.

UNIT-IV

Linear Bounded Automata: Representation of Linear Bounded Automata, String/Language Acceptability By LBA. **Turing Machine:** Representation to Turing Machine, String/Language Acceptability by Turing Machine, Universal Turing Machine, Decidability and Halting Problems of Turing Machine, Post-Correspondence Problem.

UNIT-V

Parsing Compiler: Parsing and its Types, First-And-Follow Method, **Top-Down Parsing:** Recursive Descent, Backtracking, Non-Backtracking, Predictive Parser/LL(1) Parser. **Bottom-Up Parsing:** Shift-Reduce, LR(0) Parsing, SLR(1) Parsing, LALR Parsing, CLR Parsing.

References:

1. "Theory of Computer Science: Automata, Languages and Computation" by K L P Mishra
2. "Introduction to Automata Theory, Languages, and Computation" by John E Hopcroft and Rajeev Motwani
3. "Introduction to Formal Languages and Automata" by Peter Linz, Jones & Bartlett, 2016
4. "Introduction to the Theory of Computation" by Michael Sipser
5. "Compilers Principles, Techniques and Tools" by Alfred V. Aho, Ravi Sethi, D. Jeffrey Ullman and Monica S

SOFTWARE ENGINEERING

[**PAPER CODE**]

LTP: |3|1|0|

OBJECTIVE: *The basic objective of this course is to divert the mindset of students towards the developmental process of computer software.*

OUTCOME: *Upon completion of this course the student should be able to understand the step by step process for developing software system or module.*

PREREQUISITE: *Basic knowledge of programming and project management.*

UNIT-I

Software Concepts: Introduction to Software, Characteristics, Components of System, SDLC, The Role of System Analyst, Legacy Software, The Software Crisis, Principles of Software Engineering. Requirement Analysis, Requirement Analysis Tasks, Software Prototyping.

UNIT-II

Process Models: The Process of Software Development (Waterfall, Incremental, Spiral, Concurrent Development) Agility, Agile Process Models. **Software Project Management:** Objectives, Resources and Their Estimation, LOC and FP Estimation, Effort Estimation, COCOMO Estimation Model, Risk Analysis.

UNIT-III

Software Design: Principles, Abstraction, Modularity, Software Architecture, Cohesion and Coupling, Refactoring, Structured Analysis, Evolution of Object Models. **Unified Modeling Language:** An Introduction, Views and Diagrams, UML Tools (VISIO, Lucidchart, Gliffy).

UNIT-IV

Software Testing: Software Testing, Functional and Non- Functional Testing: White Box and Black Box Testing, **Static Testing Strategies:** Static, Structural, Desk Checking, Code Walk Through, Beta, Stress, Code Inspection, Code Coverage, Code Complexity, Statement, Path, Condition, Function Coverage, Cyclomatic Complexity, Compatibility, Integration, Acceptability, Testing Tools (Selenium, Ranorex, watir).

UNIT-V

Software Quality Assurance: SQA Concepts, Garvin's Quality Dimensions, McCall's Quality Factors, Software Reviews, Software Reliability, Software Quality Standards (ISO 9000 Model, SEI CMM Model). **Software Maintenance:** Need, Categories, Problems & Cost Factors, Software Re-engineering, Reverse Engineering.

References:

1. "An Integrated Approach to Software Engineering", Pankaj Jalote, IIIrd Edition, Narosa Publishing House.
2. "Software Engineering: A Practitioner's Approach", Roger S. Pressman, McGraw-Hill.
3. "Software Engineering", Pressman, McGraw-Hill.
4. "Fundamentals of Software Engineering", Rajib Mall, Phi Learning.
5. "Software Engineering", K.K. Aggarwal, New Age.

MASTER OF COMPUTER APPLICATION
[MCA]
FOURTH SEMESTER

PAPER CODE	PAPER NAME	#L	#T	#P	TOTAL CREDIT
	Practical				
44CA471	Seminar and Presentation/Review of Literature	0	0	2	2
44CA472	Major Project-Internship	0	0	12	12
	TOTAL CREDIT	0	0	14	14