

# SYLLABUS

MANONMANIAM SUNDARANAR UNIVERSITY, TIRUNELVELI-12

**PG - COURSES – AFFILIATED COLLEGES**

Course Structure for

## **MASTER OF COMPUTER APPLICATIONS (MCA)**

(Choice Based Credit System)

(with effect from the academic year 2021- 2022 onwards)

<b>Semester-I</b>				
<b>Part</b>	<b>Subject Status</b>	<b>Subject Title</b>	<b>Subject Code</b>	<b>Credit</b>
<b>III</b>	Core - 1	Mathematical Foundations for Computer Science	ZCAM11	4
	Core - 2	Computer Organization and Architecture	ZCAM12	4
	Core - 3	Design and Analysis of Algorithms Using C++	ZCAM13	4
	Core - 4	Advanced Java Programming	ZCAM14	4
	Core - 5	Object Oriented Analysis and Design Using UML	ZCAM15	4
	Core - 6 Practical - 1	Design and Analysis of Algorithms Using C++ - Lab	ZCAL11	2
	Core - 7 Practical - 2	Advanced Java Programming - Lab	ZCAL12	2

## **MATHEMATICAL FOUNDATIONS FOR COMPUTER SCIENCE**

### **OBJECTIVES**

- To provide the strong mathematical foundation that will help the students in writing programs.
- To improve the logical reasoning while programming with computer languages.

### **OUTCOMES**

- Apply the fundamentals of set theory and matrices for the given problem.
- Apply the types of distribution, evaluate the mean and variance for the given case study/ problem.
- solve the given problem by applying the Mathematical logic concepts
- Model the given problem by applying the concepts of graph theory.



- Identify and list the different applications of discrete mathematical concepts in computer science.

## **UNIT – I**

### **SET THEORY AND MATRICES**

Sets, Operations on sets, Cardinality of sets, inclusion-exclusion principle, pigeonhole Principle, matrices, finding Eigen values and Eigen vectors.

## **UNIT – II**

### **MATHEMATICAL LOGIC**

Propositional Logic, Applications of Propositional Logic, Propositional Equivalences Predicates and Quantifiers, Nested Quantifiers, Rules of Inference Introduction to Proofs

## **UNIT – III**

### **RELATIONS**

Relations and Their Properties, n-ary Relations and Their Application, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings

## **UNIT – IV**

### **RANDOM VARIABLE AND PROBABILITY DISTRIBUTION**

Concept of random variable, discrete probability distributions, continuous probability Distributions, Mean, variance and co-variance and co-variance of random variables. Binomial And normal distribution, Exponential and normal distribution with mean and variables and Problems.

## **UNIT – V**

### **GRAPH THEORY**

Graphs and Graphs models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring

## **REFERENCES**

1. Kenneth H Rosen, “Discrete Mathematics and its Applications”, McGraw Hill Publications, 8th edition, 2018
2. Wolpole Myers Ye “Probability and Statistics for engineers and Scientist” Pearson Education, 9th edition, 2017
3. Narasingh Deo, — Graph Theory with Applications to Engineering and Computer Sciencell, Prentice-Hall of India Private Limited, 2017.



# COMPUTER ORGANIZATION AND ARCHITECTURE

## OBJECTIVES

- To understand the basic structure and operation of digital computer.
- To understand the basic processing concepts and bus organization.
- To understand the two types of control unit techniques and the concept of pipelining.
- To understand the different ways of communication with I/O devices and standard I/O interfaces

## OUTCOMES

- Understand the functional units of a computer, bus structures and addressing.
- Analyze RAM, ROM, and cache memory and virtual memory concepts.
- Evaluate the modes.
- Know about single bus, multiple bus organization.
- Design and analyze the pipelining concepts and various I/O interfaces.

## UNIT I

**Introduction:** Digital Computers-Basic Organization of a Computer-Historical Perspective.

**Digital Logic Circuits:** Digital Computers-Logic Gates-Boolean Algebra-Map Simplification-Combinational Circuits – flipflops.

**Digital Components:** Integrated Circuits – Decoders – Multiplexers – Registers-Shift Registers-Binary Counters.

**Data Representation:** Data Types – Complements-Fixed-Point Representation-Conversion of Fractions-Floating-Point Representation.

## UNIT II

### **Register Transfer and Microoperations:**

Register Transfer-Bus and Memory Transfers-Arithmetic Microoperations-Logic Microoperations-Shift Microoperations-Arithmetic Logic Shift Unit.

**Basic Computer Organization and Design:** Instruction Codes-Computer Registers-Computer Instructions-Timing and Control-Instruction Cycle.

**Programming the Basic Computer:** Machine Language-Assembly Language-The Assembler-Subroutines.

## UNIT III

**Microprogrammed Control:** Control Memory-Address Sequencing-Design of Control Unit.

**Central Processing Unit:** General Register Organization-Stack Organization-Instruction Formats-Addressing Modes-Data Transfer and Manipulation-Program Control-CISC Characteristics-RISC Characteristics.

**Pipeline and Vector Processing:** Parallel Processing – Pipelining – Arithmetic Pipeline – Instruction – RISC Pipeline – Vector processing – Array Processor.



## UNIT IV

### Computer Arithmetic:

Addition and Subtraction-Multiplication Algorithms-Division Algorithms-Decimal Arithmetic Unit-Decimal Arithmetic Operations.

**Input-Output Organization:** Peripheral Devices-Input-Output Interface-Asynchronous Data Transfer-Modes of Transfer-Direct Memory Access (DMA)-Serial Communication-Bus Standards.

## UNIT V

### Memory Organization:

Memory Hierarchy-Main Memory-Auxiliary Memory-Associative Memory-Cache Memory-Virtual Memory.

**Multiprocessors:** Characteristics of Multiprocessors-Interconnection Structures.

### Reference Books:

1. M. Morris Mano, Rajib Mall, “Computer System Architecture– Revised”, Third Edition, Pearson India Education Services Pvt. Ltd., 2017.
2. John P Heys, “Computer Architecture and Organization”, 3rd Edition, McGraw Hill 1998.
3. William Stallings, “Computer Organization and Architecture – Designing for Performance”, 10th Edition, Pearson Education, 2016.
4. Alka Viswa, “Computer Organization and Architecture”, Dreamtech Press, 2019.



# DESIGN AND ANALYSIS OF ALGORITHMS USING C++

## **OBJECTIVES**

- To learn effective problem solving in computing applications and analyze the algorithmic procedure to determine the computational complexity of algorithms.

## **OUTCOMES**

- It gives stepwise procedure to solve problems.
- The Problems can be broken down into small pieces for program development.
- Efficient approach of solving problems by a model of computations

## **UNIT - I**

### **Introduction:**

Algorithm-Specification-Performance Analysis. Data Structures: Lists – Stacks- Queues – Trees – Graphs – Dictionaries - Priority Queues.

## **UNIT - II**

### **Divide And Conquer:**

General Method-Binary Search- Finding the Maximum And Minimum- Quicksort - Strassen's Matrix Multiplication.

## **UNIT - III**

### **The Greedy Method:**

General Method- 0/1 Knapsack Problem-Job Sequencing with Deadlines – Minimum-Cost Spanning Tree- Single-Source Shortest Paths.

**Dynamic Programming:** General Method- All-Pairs Shortest Path- 0/1 Knapsack Problem.

## **UNIT - IV**

### **Basic Traversal and Search Techniques:**

Techniques for Binary Trees-Graphs-Connected Components and Spanning Trees.

**Backtracking:** General Method- N-Queen Problem- Hamiltonian Circuit Problem.

## **UNIT - V**

### **Branch and Bound:**

Method- Assignment Problem - Knapsack Problem.

**NP-Hard And NP-Complete Problem:** Basic Concepts-Cook's Theorem- - Job Shop Scheduling.

## **Reference Books:**

1. Sandeep Sen and Amit Kumar Design and Analysis of Algorithms: A



- contemporary perspective, Cambridge University Press, 2019.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", 2nd Edition, Universities Press (India) Private Ltd., 2008
  3. Aho, Hopcroft and Ullman, "The Design and Analysis of Computer Algorithm", Pearson Education, Delhi, 2001.
  4. S.Sridhar, "Design and Analysis of Algorithms", Oxford University Press, 2015.
  5. Basu S.K., "Design Methods and Analysis of Algorithms", PHI, 2006.
  6. M.A.Weiss, "Data Structures and algorithm Analysis in C++", Pearson Education, Asia, 2013.



# ADVANCED JAVA PROGRAMMING

## OBJECTIVES

- To learn the advanced concepts in the java language and java programming environment.
- To design webpage.
- To develop webpage using scripting,

## OUTCOMES

- Able to write java programs
- Understand the importance of JDBC
- Apply the Java programming techniques for providing the solution for the practical problems

## UNIT I

**OOP and Java:** Introduction to Java Language – Object Oriented Programming - Classes and Objects – Methods – Constructor - Overloading Methods.

**Inheritance:** Introduction – Inheritance Types – General form of subclass – subclass constructor – method overriding – abstract and final classes.

**Packages and Interfaces:** Defining a package – import statement – setting class path – Interfaces – Defining an interface – implementing interfaces

**Enumerations, Autoboxing and Annotations:** Enumerations – Type Wrappers – Autoboxing – Annotations (metadata)

## UNIT II

**Input/output :** I/O Basics – Byte Streams and Character Streams – Predefined Streams – Reading console input and writing console output – PrintWriter class – Reading and writing Files – Closing a File

**Generics and Collections:** Generics– Generics example – Generic class with two type parameters – Bounded types – Wildcard arguments – Generic method – Generic constructors – Generic interfaces

**Collections :** Collections – Collection interfaces – Collection classes – accessing a collection – Working with Maps – Arrays – Vector – Stack – Dictionary – Hashtable – Properties – Using Store ( ) and load ( ).

## UNIT III

**Swing :** Introduction – Components and Containers – Swing packages – simple Swing application – Event handling – JApplet – Painting in Swing – JLabel – JTextField – JButton – JCheckbox – JRadioButton – JTabbedPane – JScrollPane – JList – JComboBox – Dialogs – JTable – JMenu

**JDBC:** Java database connectivity, Types of JDBC drivers, Writing JDBC applications – Types of statement objects (Statement, PreparedStatement, and CallableStatement), Types of Resultset – Inserting and updating records – Using Transactions.



**UNIT IV**

**NIO:** NIO classes – Buffers – Channels – Path Interface – Files class – Paths class – File attribute interfaces – Reading and writing a File via a Channel – Stream-Based I/O – Path and File System operations

**Networking:** Networking Classes and Interfaces – InetAddress – TCP/IP – UDP

**RMI: Introduction** – Remote Interface – java.rmi package – Naming Class – RMI Exception – Creating a simple RMI Client/Server application.

**UNIT V**

**Java Beans:** An overview of Java Beans – Properties, Events and Methods – Introspection - Builder Tools – JDK – Persistence – Java Beans API – A Bean Example

**Java Servlets :**Java Servlets and CGI Programming – Life cycle of a Servlet – Tomcat – A Simple Servlet – javax.servlet - javax.servlet.http – Handling HTTP requests and responses – Working with Cookies – Tracking Sessions.

**References:**

1. Herbert Schildt: Java – The Complete Reference, Eleventh Edition, Tata McGraw-Hill, 2019
2. Balagurusamy : Programming with Java, Sixth Edition, Tata McGraw-Hill, 2019
3. Jim Keogh: J2EE – The Complete Reference, Second Edition, Tata McGraw-Hill, 2007
4. Patrick Niemeyer and Daniel Leuck : Learning Java, Fourth Edition, O'Reilly Media, 2013





# **OBJECT ORIENTED ANALYSIS AND DESIGN USING UML**

## **OBJECTIVES**

- To understand objects, classes and OO concepts
- To study the various phases of software development, models and design methodologies
- To use UML diagrams for software design
- To develop applications using UML

## **OUTCOMES**

- Analyze the Systems Development Life Cycle
- Identify the basic software requirements UML Modeling
- Apply software design with UML diagrams
- Develop applications using UML

## **UNIT I**

### **INTRODUCTION**

An overview - Object Basics - Object State and Properties – Behavior and Methods – Respond to Message – Encapsulation and Information Hiding – Class Hierarchy – Relationship and Associations – Aggregation – Advanced Topics – Object oriented system development life cycle

## **UNIT II**

### **METHODOLOGY AND UML**

Introduction – Survey – Rumbaugh ,Booch , Jacobson Methods – Patterns – Frameworks- Unified Modeling Language: Introduction - Static and dynamic Models – Modeling - UML Diagrams – UML Class Diagram – Use-case Diagram – UML Dynamic Modeling – Model Management – UML Extensibility

## **UNIT III**

### **OBJECT-ORIENTED ANALYSIS**

Identifying Use cases: Introduction - Business Object Analysis – Use case driven Object oriented analysis – Use-case Model – Documentation – Classification: Introduction – Classification Theory – Naming Classes Language - Constraint– Introduction – Object-Oriented Design Philosophy –UML Object Identifying Object relationships, Attributes and Methods: Super sub class – A part of relationships aggregation – Object responsibility – Methods for ViaNet bank objects.

## **UNIT IV**

### **OBJECT- ORIENTED DESIGN**

Object Oriented Design process and Design axioms: Axioms – corollaries – Design Patterns – Designing Classes : Class visibility – Refining attributes – Methods and protocols – Object – relational system – Multi Database System – Designing Interface Object - Macro and Micro level process – Purpose of a view layer interface.



## **UNIT V**

### **QUALITY AND TESTING**

Quality Assurance – Testing Strategies – Impact of Object Orientation on Testing – Test Plan – Test cases system usability – Usability Testing.

#### **Reference Books:**

1. Ali Bahrami, “Object Oriented Systems Development”, McGraw Hill Education(India) Private Limited, Edition 2008.
2. Jaya Mala, S. Geetha, “Object Oriented Analysis and Design using UML”, McGraw Hill Education, 2013
3. Atul Kahate, “Object Oriented Analysis and Design”, Tata McGraw Hill Education, 2004.
4. Mahesh P.Matha, “Object-Oriented Analysis and Design Using UML”, PHI Learning Private Limited, 2012.
5. F. Margret Sharmila, N.Jayanthi, Dr.R.Vasanthi, A.Surya, Dr.R.Palson Kennedy, “Object Oriented Analysis & Design”, Charulatha Publications Private Limited, 2019



## DESIGN AND ANALYSIS OF ALGORITHMS USING C++ - LAB

**Programs should include but not limited to:**

1. Write a program that implements Stack
2. Write a program to implement Queue
3. Write a program that implements Singly Linked List.
4. Write a program to implement Evaluation of Postfix expression.
5. Write a program to implement Priority Queue.
6. Write a program that implements Binary Tree Traversals.
7. Write a program that implements BFS
8. Write a program that implements DFS
9. Write a program to implement Binary Search using DAC technique.
10. Write a program to implement Quick Sort using DAC technique.
11. Write a program that implements Strassen's matrix multiplication using Greedy Method.
12. Write a program that implements Knapsack problem using Greedy Method.
13. Write a program that implements Prim's Algorithm
14. Write a program that implements Kruskal's Algorithm
15. Write a program that implements All-pairs Shortest path problem
16. Write a program that implements N-Queen Problem



## **ADVANCED JAVA PROGRAMMING - LAB**

1. Write a Java application program to demonstrate class with constructors and method overloading.
2. Write a Java application program to demonstrate inheritance and method overriding.
3. Write a Java application program to demonstrate package with classes and interfaces.
4. Write a Java application program to read and write a file using file I/O.
5. Write a Java application program to find minimum and maximum of integers and characters using Generics.
6. Write a Java application program to implement the Queue operations using the Linked List Class.
7. Write a Java program using the swing components with event handling.
8. Write a Java program using menus in swing.
9. Write a program in Java to create a table and insert and query records in the table.
10. Write a program in Java to read a file using Channel I/O.
11. Write a program in Java to implement a client/server environment using TCP/IP.
12. Write a program in Java to implement a client/server environment using UDP.
13. Write a program in Java to implement a client/server using RMI.
14. Write a program in Java to create a form and validate a password using Servlet.
15. Write a program in Java to develop a simple Java Bean.

