

Dr. RamManohar Lohia Avadh University, Ayodhya U.P.

Study and Evaluation Scheme

MCA (Master of Computer Applications)

(Effective From Session 2020-21)

Year – I Semester – I

MCA 103 – **Object oriented system with C++**

Course Details

- **Paper No.: III**
- **Course Code: MCA 103**
- **Course Title: Object oriented system with C++ (Theory)**
- **Marks Distribution: Internal = 50, External = 100, Total = 150**

Goal

1. **To learn the methodologies of object-oriented analysis and design.**
2. **To advance programming knowledge through C++ with emphasis on OOP principles, data abstraction, inheritance, polymorphism, templates, operator overloading, and exception handling.**

Outcomes

At the end of the course, the student should be able to:

1. Acquire knowledge of **OOAD**.
2. Demonstrate **design** concepts using **UML diagrams**.
3. Apply the **object-oriented life cycle**.
4. Draw **UML** diagrams.
5. Design applications using **OOAD tools**.
6. Explain **object-oriented programming** concepts.
7. Write simple programs in **C++**.
8. Demonstrate functions, operator overloading, and inheritance in C++.
9. Implement exception handling, generic functions, and templates.

UNIT I — Introduction to Object Oriented Systems & Methodologies

- Overview of object-oriented systems development, Object Basics
- Object-Oriented Systems Development Life Cycle
- Object-Oriented Methodologies: Rumbaugh, Booch, Jacobson
- Patterns, Frameworks, Unified Approach, UML Diagrams: Use case, Class, Interaction, Package, Collaboration, State, Activity
- Object-Oriented Analysis: Identifying use cases, Classification, Object relationships, Attributes & Methods
- Object-Oriented Design: Design axioms, Class design, Access Layer, Object Storage, Interoperability

UNIT II — Object & Dynamic Modeling

- Object Modeling: Objects & classes, Links, Associations, Generalization, Inheritance, Aggregation, Abstract classes, Multiple inheritance, Metadata, Candidate keys, Constraints
- Dynamic Modeling: Events & States, Operations & Methods, Nested State Diagrams, Concurrency, Advanced dynamic model concepts

UNIT III — Functional Modeling & Design Implementation

- Functional Modeling: Functional models, Data flow diagrams, Specifying operations, Constraints
- Translating OOD into implementation
- OMT methodologies, Case studies, Methodology comparison: OOSD, SA/SD, JSD
- Application design using OOSD tools

UNIT IV — Programming in C++ & Functions

- Introduction to C++ OOP paradigm, Programming paradigms comparison
- Structure of a C++ program, Difference between C & C++
- Variables, constants, enumerations, arrays, pointers, references
- Functions & overloading: Constructors, Copy constructor, Assignment operator, Access modifiers
- Operator overloading: Unary, Binary, Arithmetic, Comparison, Assignment, Subscript, Function call, Input/Output
- Namespaces, Compile-time polymorphism

UNIT V — Inheritance, Polymorphism, Exception Handling & Templates

- Inheritance: Types, Abstract classes, Overriding, Constructors & Destructors, Multiple inheritance
- Polymorphism: Compile-time & run-time, Dynamic polymorphism, Pointers to objects, Virtual functions, Pure virtual functions, Abstract class
- Exception Handling: Implementation, Advanced I/O, Manipulating strings, Stream manipulators
- Files: Reading and writing objects, File I/O operations
- Templates: Generic programming, Function templates, Standard Template Library (STL), Containers, Iterators, Algorithms, Sequence & Associative containers