

## Assignment 2 : Object-Oriented Programming (OOP) in C++

**Subject: Object-Oriented Programming (OOP)**

**Instructor: Rahul Rawat**

**Student Name: \_\_Shaurya Agarwal\_\_\_\_\_**

**Date: \_\_04-08-2025\_\_\_\_\_**

### Instructions:

- Answer all questions clearly.
- Use appropriate C++ syntax where code is asked.
- Write neatly and attach extra sheets if needed.
- This is a open book assignment so you are allowed to revise topic before attempting each question.

### Part A: Theory Questions

1. Define the following terms with suitable examples:
  - a) Class
  - b) Object
  - c) Data Members
  - d) Member Functions
2. Differentiate between:
  - a) Class and Object
  - b) Getter and Setter
  - c) Constructor and Destructor
  - d) Public and Private access modifiers
3. Explain the concept of a constructor in C++. What are its types? Explain with examples:
  - a) Default Constructor
  - b) Parameterized Constructor
  - c) Copy Constructor
4. What is a destructor? When is it invoked in C++? Provide an example.
5. Explain the use of "this" keyword in C++. Give a example where it is necessary.

## Part B: Practical Questions

### 6. Write a C++ program to define a "Student" class with the following:

- Data Members:
  - i. "rollNumber"
  - ii. "name"
  - iii. "marks"
- Member Functions:
  - i. "setDetails()" – to input data in all the data members using setter method
  - ii. "displayDetails()" – to display all student information
- Implement:
  - i. Three different Constructor to initialize values(Use "this" pointer)
  - ii. Getter and Setter methods

### 7. Create a class "Rectangle" with:

- Private data members:
  - i. "length"
  - ii. "breadth"
- Public functions:
  - i. Constructor (both default and parameterized)
  - ii. Destructor
  - iii. "area()" function to return area
  - iv. "perimeter()" function
  - v. Getter and Setter methods for both length and breadth

## Bonus Question :

### 8. Write a program in C++ that demonstrates all types of access modifiers ("public", "private", and "protected") in a single class or using inheritance.

### 9. Library System –

- Create a C++ program that models a Library System using inheritance:
- Create a base class "LibraryItem" with data members: "title", "itemID", and a method "displayInfo()".
- Derive a class "Book" from "LibraryItem" with additional members: "author", "pageCount".
- Create another derived class "DVD" from "LibraryItem" with additional members: "duration" (in minutes), "genre".
- In "main()", create one object(both static and dynamic) of each derived class, initialize data using constructors or setters, and call "displayInfo()".

- Goal: Understand how common attributes can be inherited while allowing specific details for different item types.

## **10. University System –**

- Write a C++ program to simulate a University System that demonstrates the diamond problem and how to solve it using virtual inheritance:
- Create a base class “UniversityMember” with data members: “name”, “id”, and a function “displayDetails()”.
- Derive two classes:
  1. “Teacher” (virtually inherits from “UniversityMember”)
  2. “Researcher” (virtually inherits from “UniversityMember”)
  3. Now create a class “TeachingAssistant” that inherits from both “Teacher” and “Researcher”.
- In “main()”, create an object of “TeachingAssistant”, set the values, and call the “displayDetails()” function to show that there is no ambiguity.
- Goal: Demonstrate that using virtual inheritance avoids multiple copies of the base class.

**Submission Deadline: \_\_10-08-2025\_\_\_\_\_**