

# Assignment Programs

## 1. Create a Class and Object

- o Define a `Student` class with attributes (`name`, `rollNo`), create objects and display details.

## 2. Constructor Example

- o Create a `Book` class with a constructor to initialize book name and author, and a method to display them.

## 3. Default and Parameterized Constructor

- o `Car` class with two constructors: one default and one parameterized.

## 4. Function Overloading

- o `Calculator` class with multiple `add()` methods:
  - `add(int, int)`, `add(double, double)`, `add(int, int, int)`

## 5. Constructor Overloading

- o `Employee` class with overloaded constructors to initialize with different sets of data (e.g., name only, name and id, name, id, and salary).

## 6. Class with Method to Calculate Area

- o Create a `Rectangle` class with length and width, and a method `calculateArea()`.

## 7. Student Class with Marks and Average

- o Accept marks of 3 subjects using constructor, calculate average using method.

## 8. Bank Account Class

- o Class `BankAccount` with deposit, withdraw, and showBalance methods; use constructors to initialize account.

## 9. Class with Object as a Member

- o Create `Address` and `Employee` classes. `Employee` has an `Address` object as a member.

## 10. Function Overloading in Constructor and Method

- `Shape` class with overloaded constructors for circle and rectangle. Also overload `area()` method to handle both shapes.

## 11. Class with Private Members and Public Getters/Setters

- `Student` class with private fields (`name`, `age`) and public methods to access them using getter/setter methods. Use constructor to initialize.

## 12. Array of Objects

- Create a `Product` class and an array of `Product` objects. Accept data and display all products using loop.

## 13. Constructor with Validation using Exception

- `Employee` constructor throws an exception if salary is negative.

## 14. Custom Exception Handling

- Create a custom exception `InvalidAgeException`. Throw it if `age < 18` in a method `checkEligibility()`.

## 15. Static vs Non-static Members

- `University` class with static `universityName` and non-static `studentName`. Demonstrate calling static vs non-static members.

## 16. Multiple Classes with Relationships

- `Department` and `Professor` class. Each `Professor` is linked to a `Department` object.

## 17. Array of Objects with Total Calculation

- `Marks` class having subject marks, use array of students to calculate and display total and average marks.

## 18. Banking System with Exception and Access Modifiers

- Create a `BankAccount` class with private `balance`, public `deposit()` and `withdraw()`. Throw exception if withdrawal amount > balance.

## 19. Constructor Calling Another Constructor (`this()`)

- Use `this()` to chain constructors inside a `Customer` class.

## 20. Library Management with Object Array and Search

- `Book` class with ID, title, author. Store multiple books and allow searching by book title.