



C++-A15: Overloading

C++ Implementation-Based Problems

1. Function Overloading: Geometry Calculator

Problem:

- Create a class `Geometry` with an overloaded function `calculateArea()` to compute:
 - Area of a square (`calculateArea(int side)`)
 - Area of a rectangle (`calculateArea(int length, int breadth)`)
 - Area of a circle (`calculateArea(double radius)`)
 - Demonstrate function overloading in `main()` .
-

2. Function Overloading: Volume Calculation

Problem:

- Define a class `Volume` with overloaded functions `calculate()` :
 - `calculate(int side)` : Volume of a cube
 - `calculate(int length, int breadth, int height)` : Volume of a cuboid
 - `calculate(double radius, double height)` : Volume of a cylinder
 - Call all versions in `main()` .
-

3. Constructor Overloading: Bank Account Management

Problem:

- Create a class `BankAccount` with:
 - `private` members: `accountNumber` , `balance` , `accountHolder`
 - Three constructors:

1. Default constructor
 2. Constructor with `accountNumber` and `accountHolder`
 3. Constructor with all three attributes
- Demonstrate constructor overloading in `main()` .
-

4. Constructor Overloading: Student Details

Problem:

- Define a class `Student` with:
 - `private` members: `name` , `rollNo` , `grade`
 - Three constructors:
 1. Default constructor
 2. Constructor with `name` and `rollNo`
 3. Constructor with all three attributes
 - Create student objects using different constructors in `main()` .
-

5. Operator Overloading: Complex Number Addition

Problem:

- Define a class `Complex` with `private` members `real` and `imaginary` .
 - Overload the `+` operator to add two complex numbers.
 - Overload the `-` operator to subtract two complex numbers.
 - Implement `main()` to test operator overloading.
-

6. Operator Overloading: Matrix Addition and Subtraction

Problem:

- Create a class `Matrix` with:
 - A `private` 2D array to store elements
 - Overload the `+` operator to add two matrices
 - Overload the `-` operator to subtract two matrices
 - A function to display the matrix

- Demonstrate both operations in `main()` .
-

7. Operator Overloading: String Concatenation

Problem:

- Define a class `String` with a `private` character array.
 - Overload the `+` operator to concatenate two strings.
 - Overload the `==` operator to compare two strings.
 - Test string operations in `main()` .
-

8. Operator Overloading: Overloading Increment and Decrement Operators

Problem:

- Create a class `Counter` with a `private` integer `count` .
 - Overload `++` (pre-increment and post-increment) and `-` (pre-decrement and post-decrement) operators.
 - Demonstrate the operations in `main()` .
-

9. Function Overloading: Math Operations

Problem:

- Define a class `MathOperations` with overloaded `calculate()` :
 - `calculate(int, int)` : Sum of two integers
 - `calculate(double, double)` : Sum of two floating-point numbers
 - `calculate(int, double)` : Product of an integer and a floating-point number
 - Demonstrate function overloading in `main()` .
-

10. Operator Overloading: Overloading '[]' for Array Access

Problem:

- Create a class `Array` with:
 - A `private` integer array

- Overload the `[]` operator to access elements safely (with bounds checking)
 - Demonstrate operator overloading in `main()` .
-

Happy Coding!