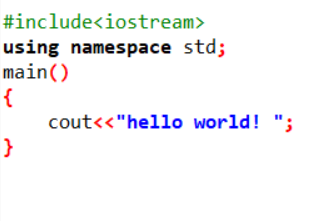
MODULE: 4

# OOPS Concept

## Topics Covered

### Basic Concepts of OOP

1. WAP to print “Hello World” using C++

 2. What is OOP? List OOP concepts

Oop: object oriented programming system.

List OOP concepts :

1.class & objects

2.inheritance

3.polymorphism

4.encapsulation

5.abstraction

3. What is the difference between OOP and POP?

OPPS:-

-> Objects as foundational elements The foundational elements of OOP are objects. Objects live within a class and help to define the information collected, stored, and used within the class. In OOP, programs are broken down into objects.

-> Use of classes A class is simply the umbrella over objects that tells the object what information to collect, store, and use. Classes are also used for “message passing” to communicate between classes in the program.

-> Bottom-up design approach Four pillars of OOP design: Abstraction: Used in OOP to simplify the user experience by hiding internal details Encapsulation: Functions by bundling related attributes and methods together Inheritance: The concept that classes and objects can take on elements of other classes and objects that are already in existence by inheriting their properties Polymorphism: Concept of using one chain for a variety of applications.

POP:-

-> Functions as foundational elements The foundational elements of POP are functions. Functions are the product of the broken-down segments within program-oriented programming. The benefits of functions are that they can change the form of any piece of data at any time and place.

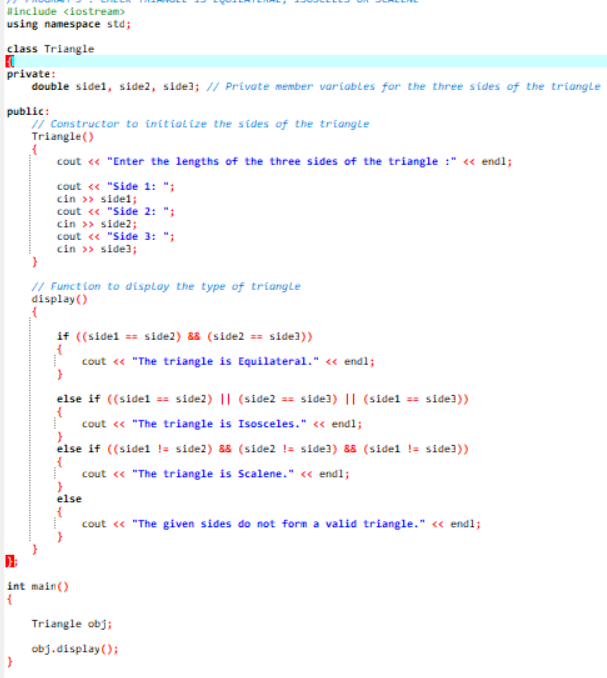
-> Use of global data Within POP, functions typically share global data. This can be useful when multiple functions need to access the same data. Because POP utilises the function of global data, data is free to move throughout the program from function to function.

-> Top-down design approach POP follows a particular procedure or order via a structured flowchart where the program completes tasks in top- down sequential order.

# Topics Covered

Basic Concepts of OOP

1. WAP to create simple calculator using class



2. Define a class to represent a bank account. Include the following members:

3. Data Member:

-Name of the depositor

-Account Number

-Type of Account

-Balance amount in the account

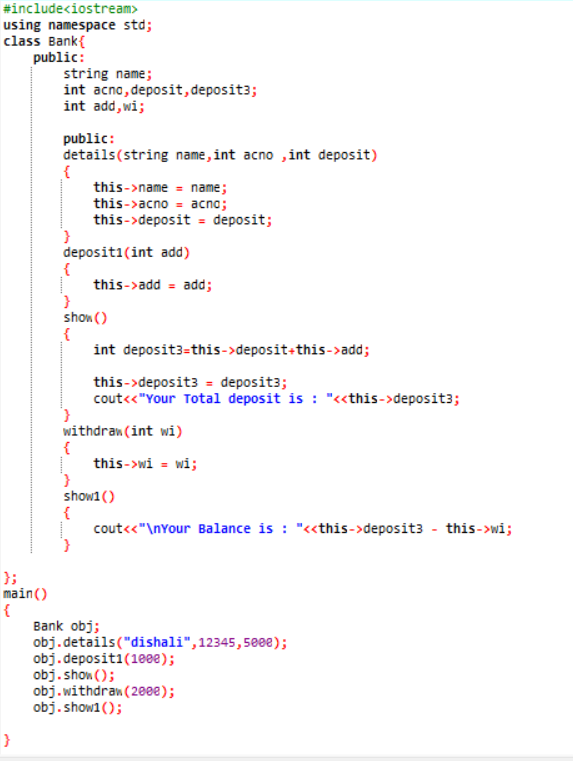
Member Functions

-To assign values

-To deposited an amount

-To withdraw an amount after checking balance

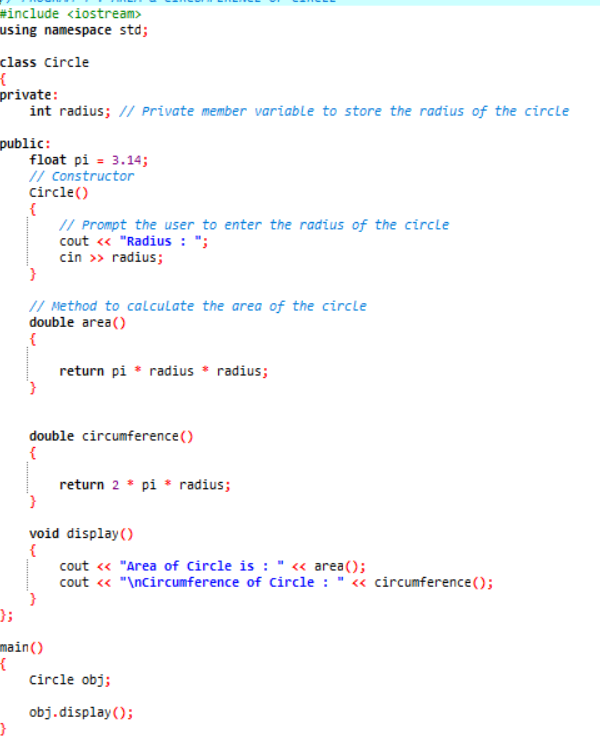
-To display name and balance



4. Write a C++ program to implement a class called Circle that has private

member variables for radius. Include member functions to calculate the

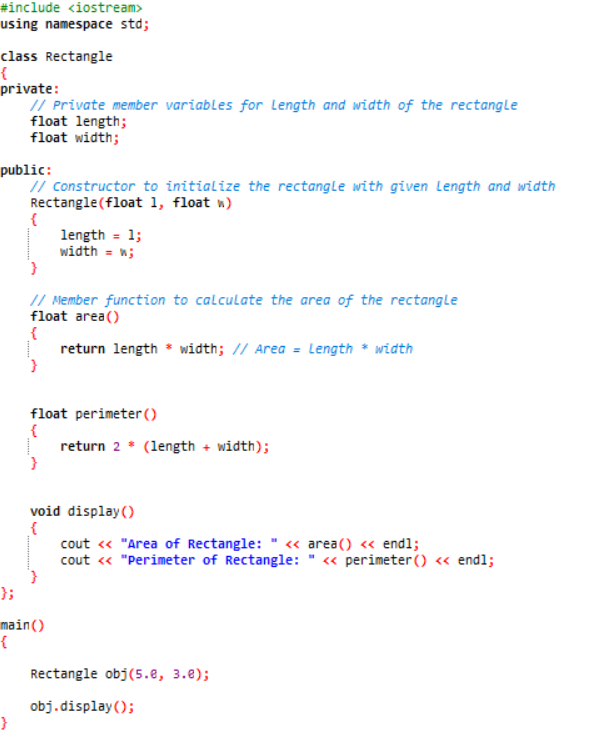
circle's area and circumference.



5. Write a C++ program to create a class called Rectangle that has private

member variables for length and width. Implement member functions to

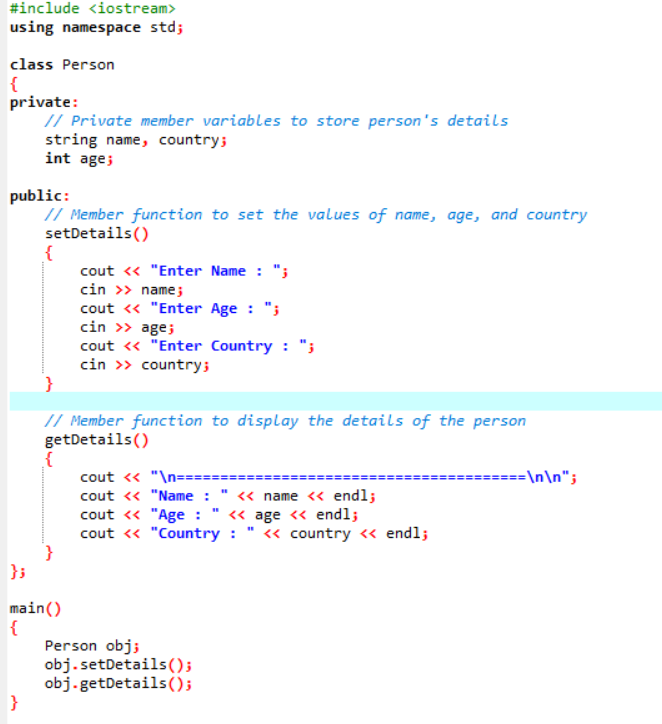
calculate the rectangle's area and perimeter.



6. Write a C++ program to create a class called Person that has private

member variables for name, age and country. Implement member

functions to set and get the values of these variables.



Topics Covered

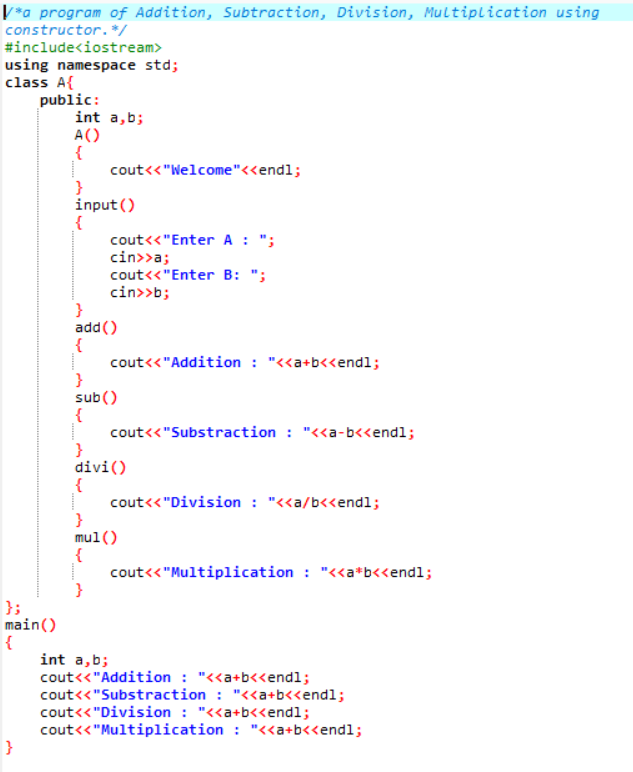
Constructor

Destructor

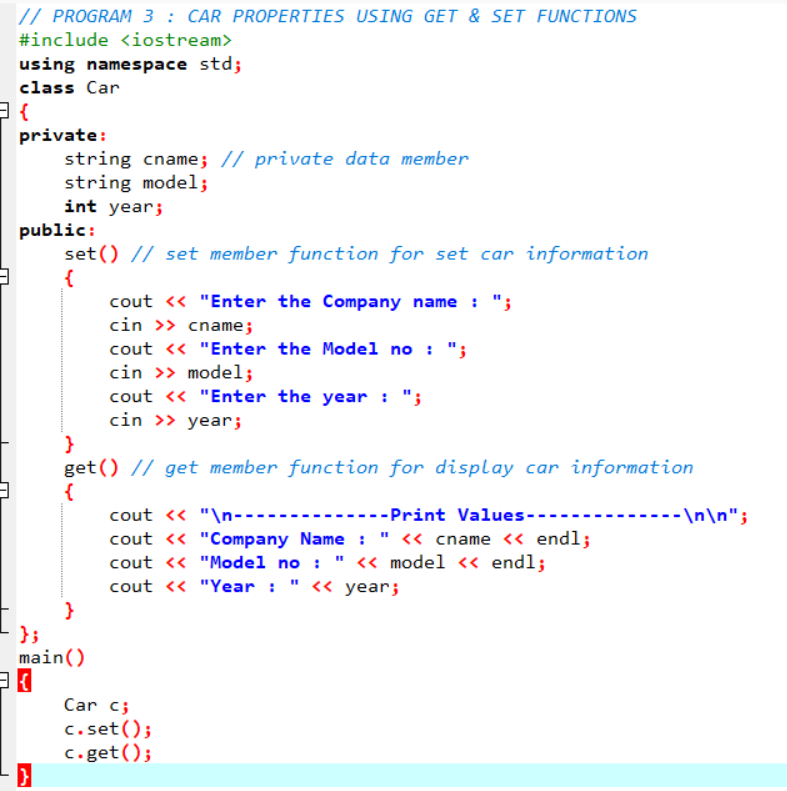
Encapsulation

Abstraction

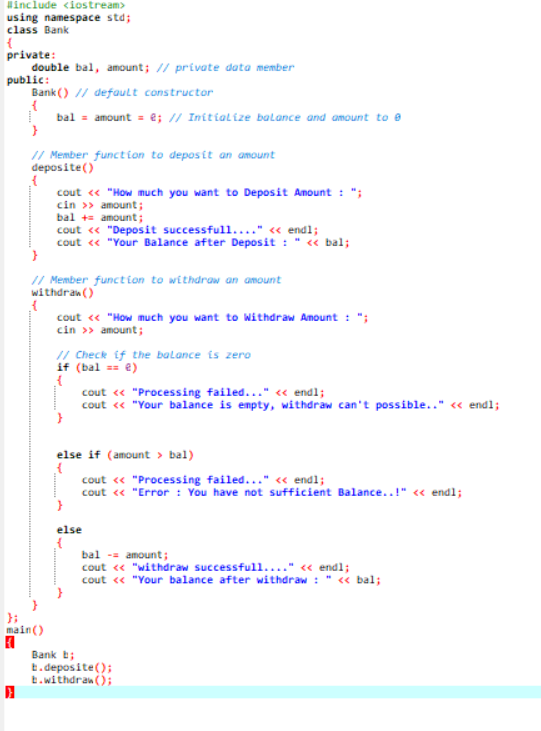
1. Write a program to find the multiplication values and the cubic values using inline function
2. Write a program of Addition, Subtraction, Division, Multiplication using constructor.



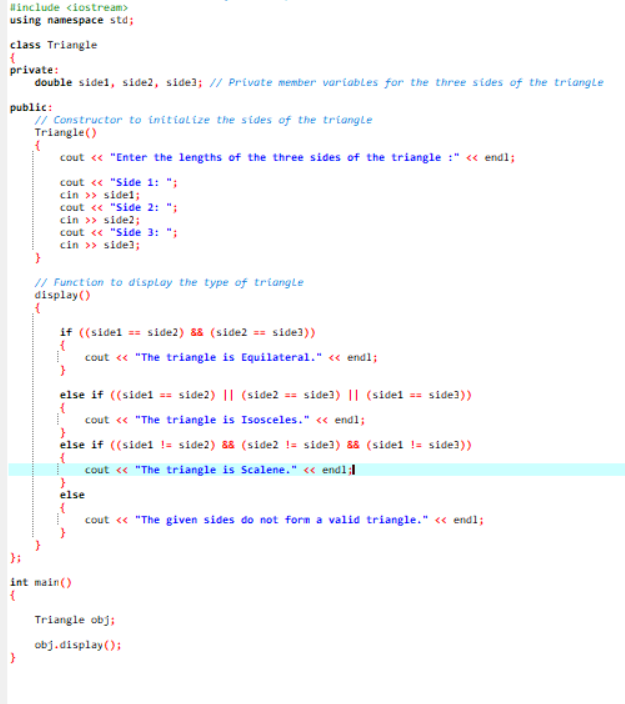
1. Write a C++ program to create a class called Car that has private member variables for company, model, and year.



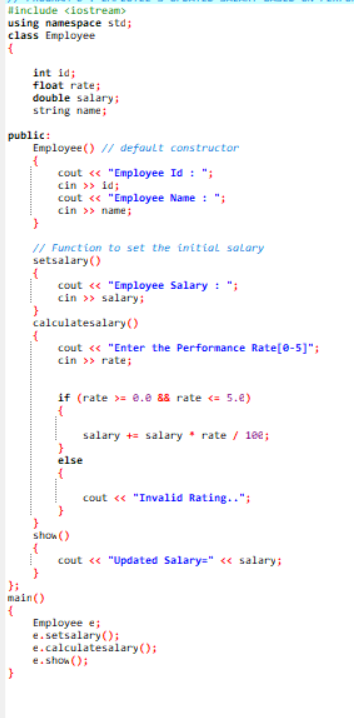
1. Implement member functions to get and set these variables. 4. Write a C++ program to implement a class called Bank Account that has private member variables for account number and balance. Include member functions to deposit and withdraw money from the account.



1. Write a C++ program to create a class called Triangle that has private member variables for the lengths of its three sides. Implement member functions to determine if the triangle is equilateral, isosceles, or scalene.

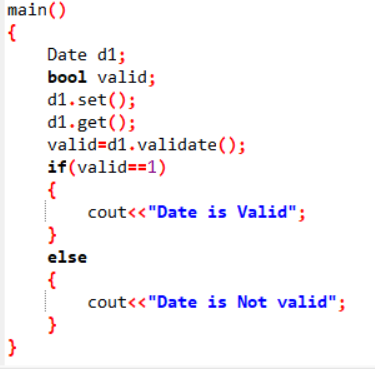


1. Write a C++ program to implement a class called Employee that has private member variables for name, employee ID, and salary. Include member functions to calculate and set salary based on employee performance. Using of constructor

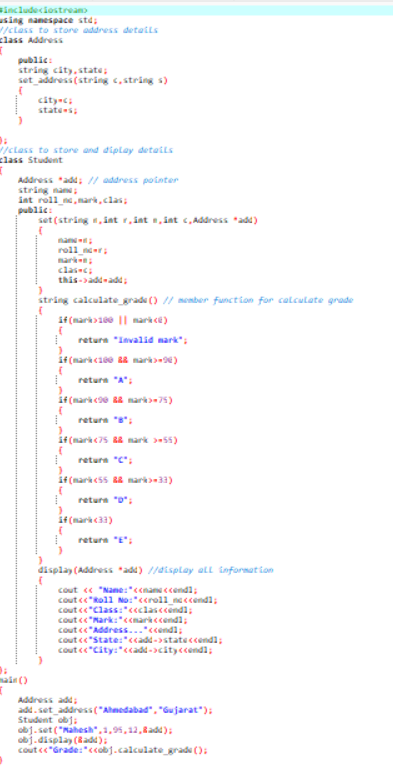


1. Write a C++ program to implement a class called Date that has private member variables for day, month, and year. Include member functions to set and get these variables, as well as to validate if the date is valid.

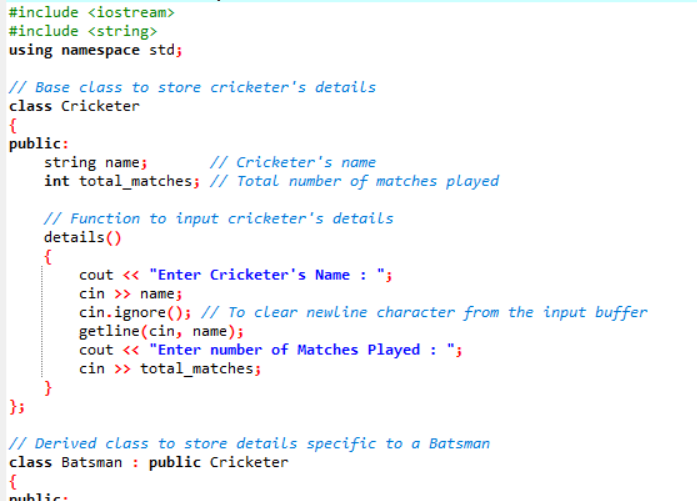


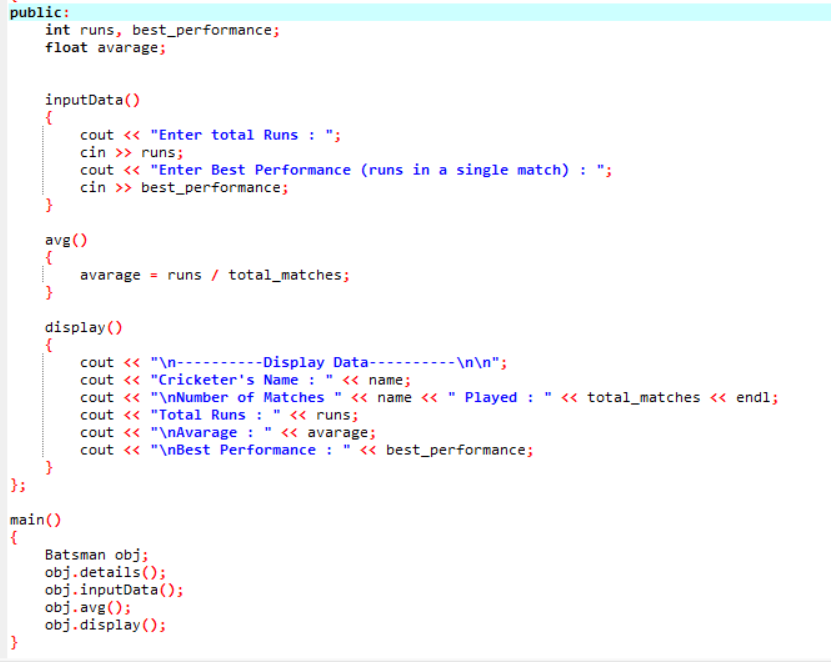


1. Write a C++ program to implement a class called Student that has private member variables for name, class, roll number, and marks. Include member functions to calculate the grade based on the marks and display the student's information. Accept address from each student implement using of aggregation

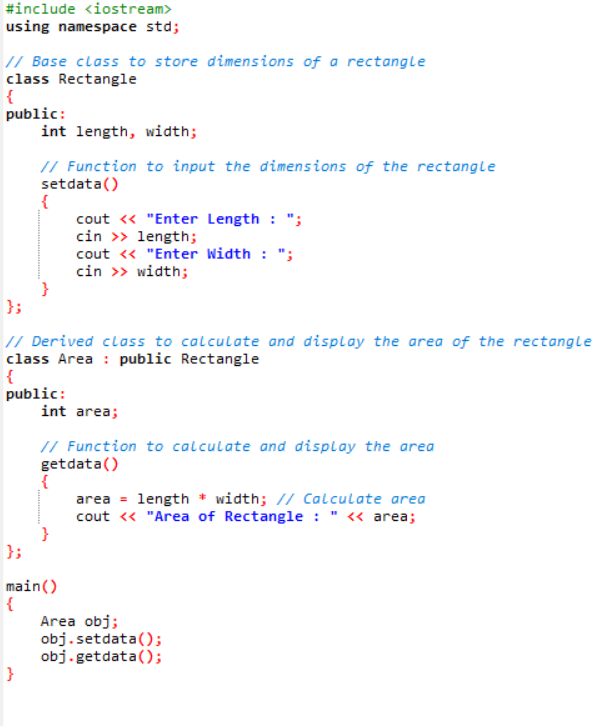


1. Assume a class cricketer is declared. Declare a derived class batsman from cricketer. Data member of batsman. Total runs, Average runs and best performance. Member functions input data, calculate average runs, Display data. (Single Inheritance)

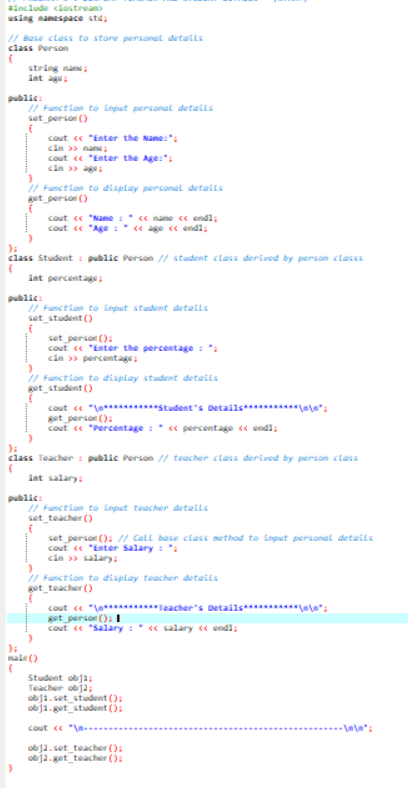




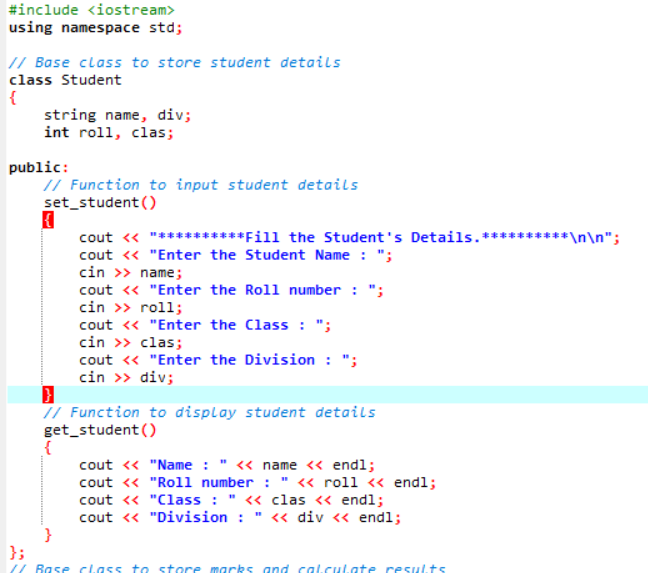
1. Write a C++ Program to find Area of Rectangle using inheritance

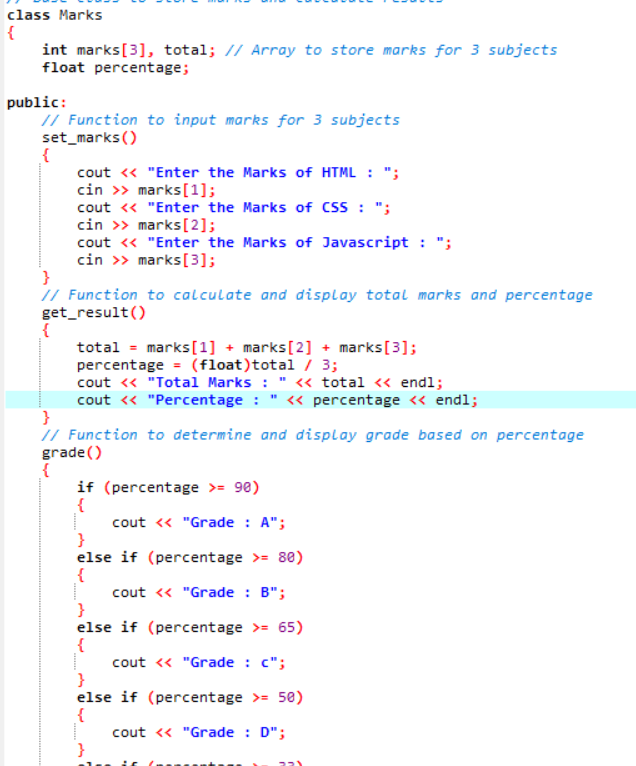


1. Create a class person having members name and age. Derive a class student having member percentage. Derive another class teacher having member salary. Write also Main function (Multiple Inheritance)



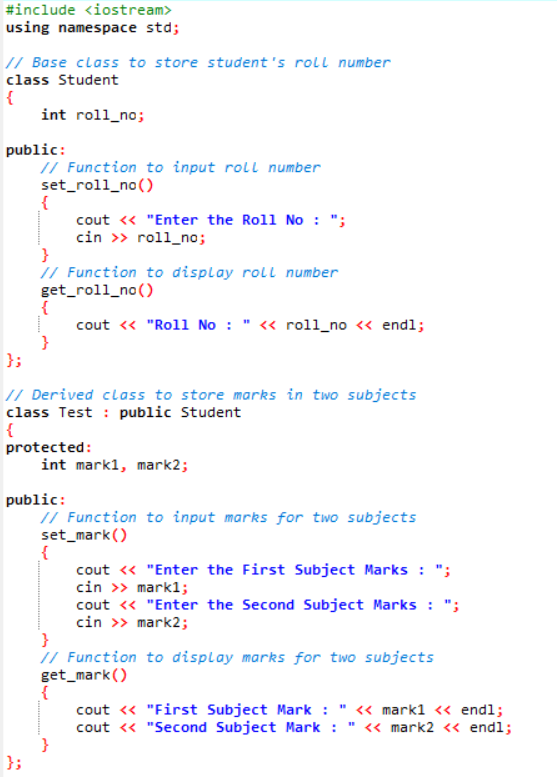
1. Write a C++ Program display Student Mark sheet using Multiple inheritance



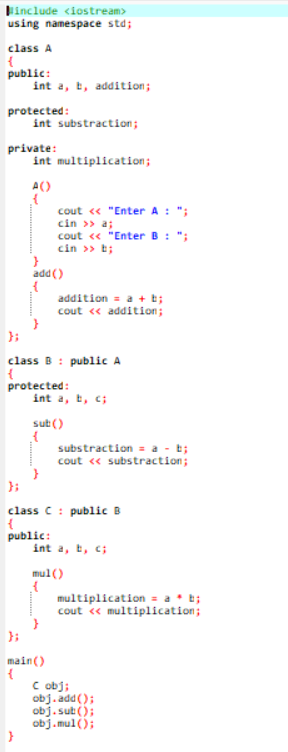




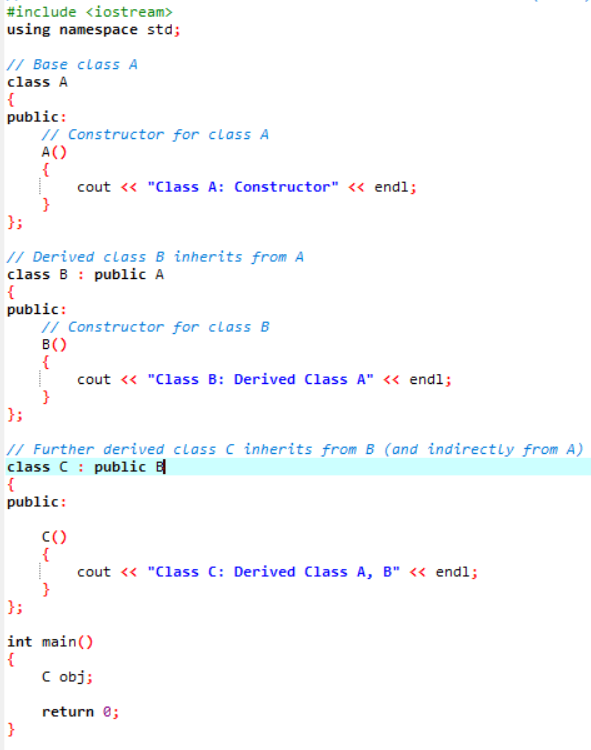
1. Assume that the test results of a batch of students are stored in three different classes. Class Students are storing the roll number. Class Test stores the marks obtained in two subjects and class result contains the total marks obtained in the test. The class result can inherit the details of the marks obtained in the test and roll number of students. (Multilevel Inheritance)



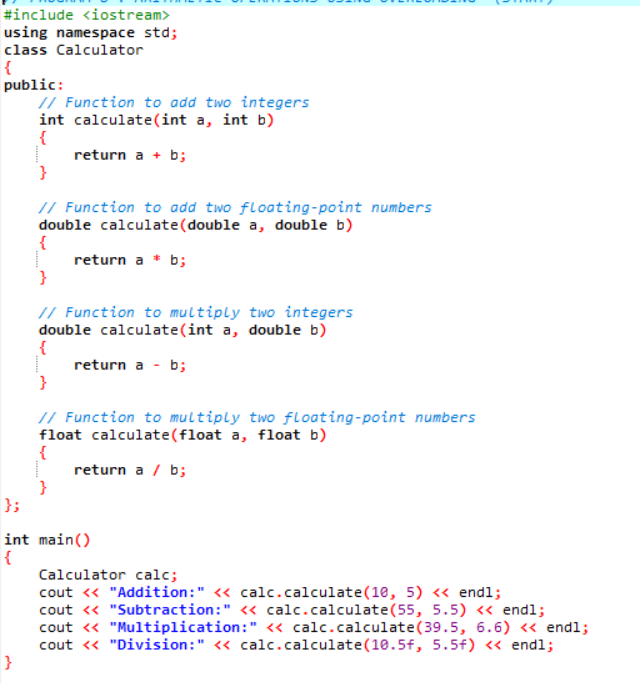
1. Write a C++ Program to show access to Private Public and Protected using Inheritance



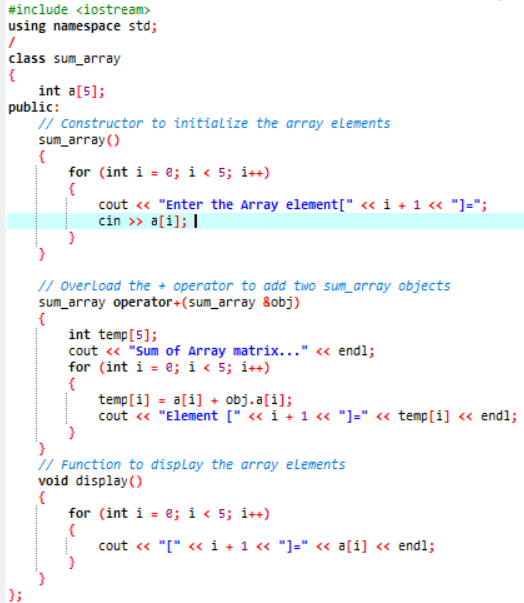
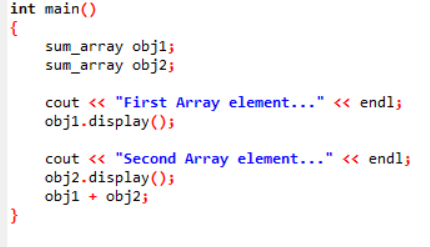
1. Write a C++ Program to illustrates the use of Constructors in multilevel inheritance



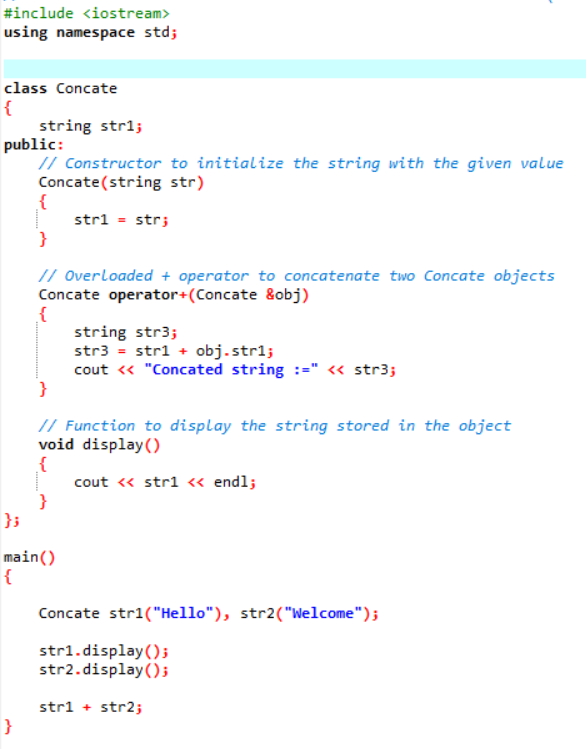
1. Write a program to Mathematic operation like Addition, Subtraction, Multiplication, Division Of two number using different parameters and Function Overloading



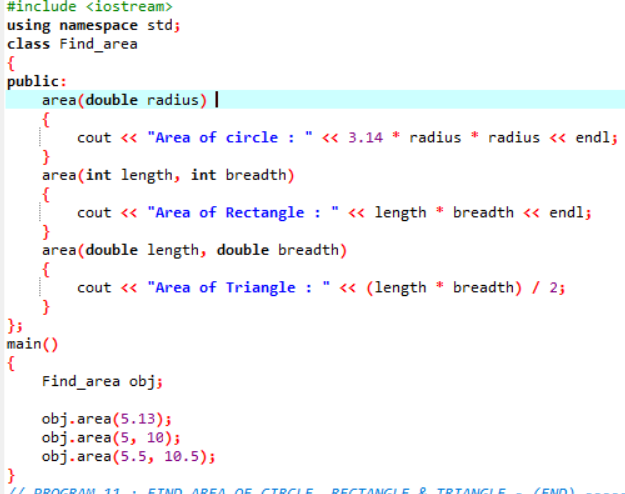
1. Write a Program of Two 1D Matrix Addition using Operator Overloading

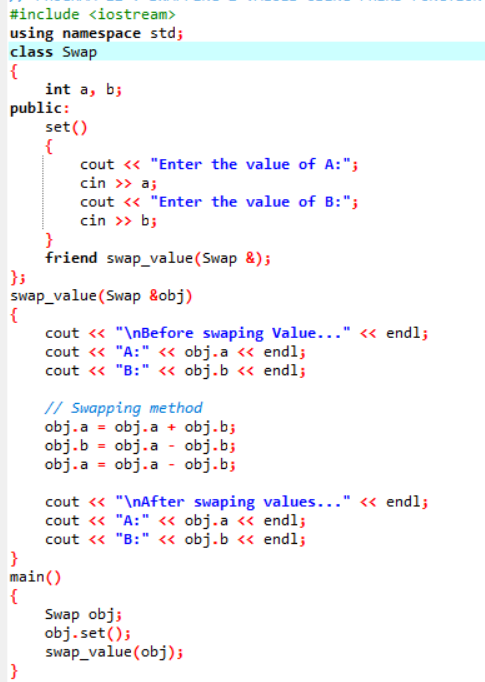
10.Write a program to concatenate the two strings using Operator Overloading



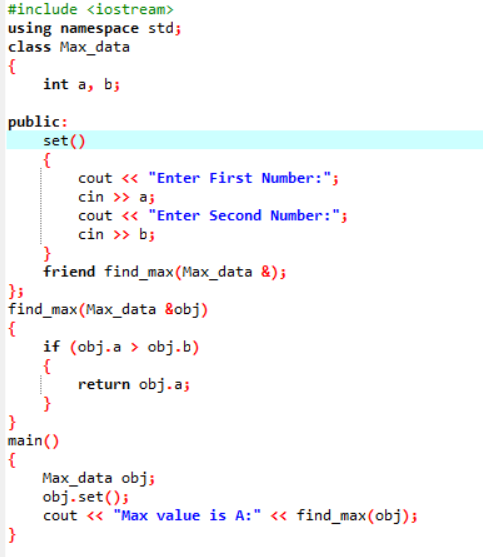
11.Write a program to calculate the area of circle, rectangle and triangle using Function Overloading Rectangle: Area \* breadth Triangle: ½ \*Area\* breadth

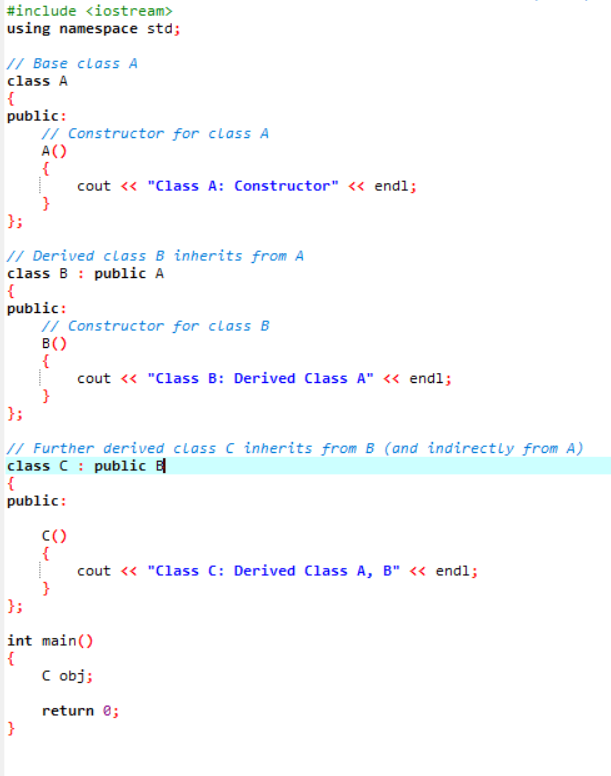


12.Write a program to swap the two numbers using friend function without using third variable



13.Write a program to find the max number from given two numbers using friend function

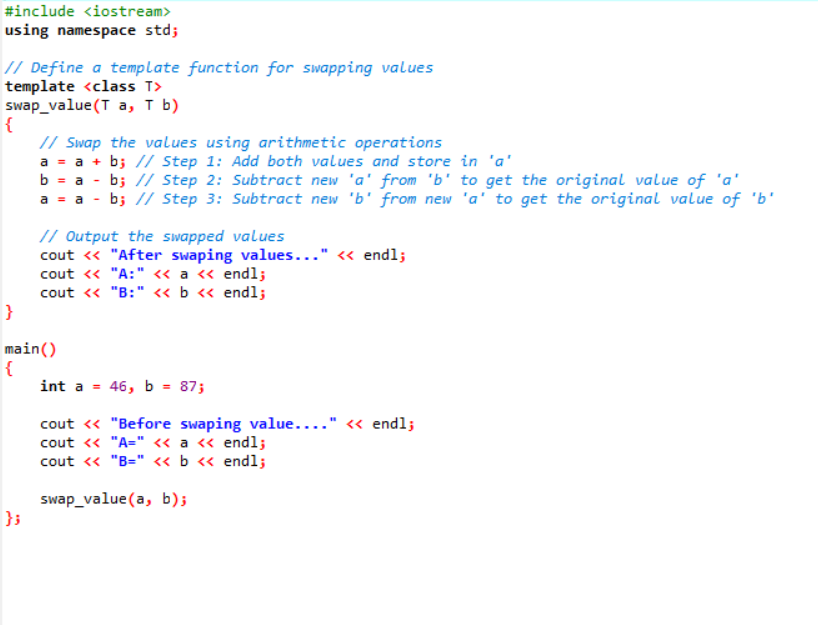


Circle: Pi \* Area \*Area

Topics Covered

Templates

1. Write a program of to swap the two values using template



1. Write a program of to sort the array using templates

