

1. Write a C++ program that prints your name, id, department, semester, and section.
2. Write a C++ program that reads two integers from the user and prints their sum.
3. Write a C++ program to find the largest of three numbers provided by the user.
4. Write a C++ program that checks whether a given integer is even or odd.
5. Write a C++ program that calculates the factorial of a given number using a loop.
6. Write a C++ program to generate the first n numbers of the Fibonacci sequence, where n is provided by the user.
7. Write a C++ program that checks whether a given number is prime.
8. Write a C++ program that calculates the sum of all elements in a given array of integers.
9. Write a C++ program that includes a function to calculate the area of a circle. The function should take the radius as a parameter and return the area.
10. Write a C++ program that performs basic arithmetic operations (addition, subtraction, multiplication, division) on two numbers provided by the user. The user should be able to choose the operation.

Problem 1: (Sum of Two Numbers) Write a function named *sum* that takes two integers as arguments and returns their sum. Use this function in your main program to add two user-input numbers and display the result.

Problem 2: (Find the Maximum of Three Numbers) Write a function named *findMax* that takes three integers as arguments and returns the maximum of the three. Call this function in the main program and print the maximum value.

Problem 3: (Check if a Number is Prime) Write a function named *isPrime* that takes an integer as an argument and returns true if the number is prime and false otherwise. Test this function with user input in your main program.

Problem 4: (Factorial of a Number) Write a function named *factorial* that takes a non-negative integer as an argument and returns its factorial. Use this function in the main program to compute and display the factorial of a user-input number.

Problem 5: (Calculate the Power of a Number) Write a function named *power* that takes two integers base and exponent as arguments and returns base raised to the power of exponent. Implement this using a loop (do not use the pow function from the standard library). Use this function in your main program.

Problem 6: Define a structure named *Person* with the following members:

- name (string)
- age (int)
- height (float)

Write a program that:

- Declares a Person variable.
- Accepts input for the name, age, and height from the user.
- Displays the information of the Person.

Problem 7: Create a class named *Rectangle* with two private data members: length and width. Include public methods to:

- Set the length and width.
- Calculate and return the area of the rectangle.
- Calculate and return the perimeter of the rectangle.
- Display the length and width of the rectangle.

Problem 8: Create a class named *Student* with the following private data members: name, rollNumber, and marks. Provide public methods to:

- Set the student's name, roll number, and marks.
- Display the student's details.
- Check if the student has passed, assuming the passing marks are 40.

- 1.** 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.
- 2.** 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.
- 3.** Write a C++ program to implement a class called BankAccount that has private member variables for account number and balance. Include member functions to deposit and withdraw money from the account.
- 4.** 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.
- 5.** 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.

CSE 1202 (Object Oriented Programming Lab)

Lab 4 - Class Constructor

1. Create a class named Car with private data members color, model, and year. Include public methods to:
 - Set the car's color, model, and year.
 - Display the car's details.
 - Check if the car is considered a vintage car (older than 25 years).
2. Create a class Box with private data members length, width, and height. Implement a default constructor to initialize these members to 0. Write a program to create an object of the Box class and display the values of its members.
3. Create a class Circle with a private data member radius. Implement a parameterized constructor that initializes radius with the given value. Write a program that calculates and displays the area of the circle.
4. Create a class Point with private data members x and y coordinates. Implement:
 - A default constructor that initializes both x and y to 0.
 - A parameterized constructor that initializes x and y with specific values. Write a program to create objects using both constructors and display the coordinates of each point.
5. Create a class Student with private data members name and grade. Implement a constructor with default arguments where:
 - name is initialized to "Unknown".
 - grade is initialized to 0. Write a program that creates multiple Student objects, some with given names and grades and others with default values.

CSE 1202

(Constructor, Destructor, and Objects as Function Parameter)

1. Create a class `Person` with private members `name` and `age`. Implement:

- A parameterized constructor to initialize `name` and `age`.
- A destructor to display a message when the object is destroyed. Write a program that creates a `Person` object and displays the details.

2. Create a class `Rectangle` with private members `length` and `width`. Implement:

- A default constructor to initialize both `length` and `width` to 1.
- A parameterized constructor to initialize `length` and `width` with given values.
- A destructor to display a message when the object is destroyed. Write a program to create two `Rectangle` objects, one with the default constructor and one with the parameterized constructor, and display their areas.

3. Create a class `Rectangle` with private members `length` and `width`. Implement:

- A parameterized constructor to initialize the dimensions.
- A method `area()` to calculate and return the area.
- A function `compareArea(Rectangle r)` that takes a `Rectangle` object by value and compares its area with the current object's area. Write a program to create two `Rectangle` objects and use the `compareArea()` function to compare their areas.

4. Create a class `Complex` to represent a complex number with real and imaginary parts. Implement:

- A parameterized constructor to initialize the complex number.
- A function `addComplex(Complex c1, Complex c2)` that takes two `Complex` objects as arguments and returns their sum as a new `Complex` object. Write a program to demonstrate adding two complex numbers using this function.

Object Oriented Programming Lab

(CSE 1202)

(Friend Function, Copy Constructor)

1. Write a class BankAccount with private members for account holder's name, account number, and balance. Use a constructor to initialize these values and a destructor to display a message indicating when the object is destroyed. Create a BankAccount object and display the account details.
2. Write a class SwapValues with two private integer members. Implement a friend function that swaps the values of the two private members and displays the swapped values.
3. Create two classes, Rectangle and Circle, each with a private member representing their area. Write a friend function that compares the areas of the two objects and prints which object has a larger area.
4. Write three classes: ClassX, ClassY, and ClassZ, each containing a private integer member. Implement a single friend function that has access to the private members of all three classes and finds the maximum value among them.
5. Define a class Student with private members for name, roll number, and marks. Implement a copy constructor to copy the details of one student object to another. Write a destructor to display a message when the object is destroyed. Test this by creating a Student object and copying it to another.