1# Write a C++ program that uses functions to perform the following operations:

- i. To insert a sub string into a given main string from a given position.
- ii. To delete n characters from a given position in a given string

```
[] G Run
                                                                                             Output
main.cpp
 1 #include <iostream>
                                                                                            /tmp/JvJkzf7zJV.o
2 #include <string>
                                                                                            After insertion: Hello, C++ World!
                                                                                            After deletion: Hello, World!
 3 using namespace std;
5 \, // Function to insert a substring into a main string at a given position
6 -
        string insert_str(string m_str, string sub_str, int position) {
        m_str.insert(position, sub_str);
       return m_str;
10
11 // Function to delete n characters from a given position in a given string
12 -
       string del_char(string m_str, int position, int n) {
13
        m_str.erase(position, n);
        return m_str;
15 }
16
17 - int main() {
      string m_str = "Hello, World!";
18
        string sub_str = "C++ ";
       int position = 7;
20
21
       // Insert the substring into the main string at the given position
string result = insert_str(m_str, sub_str, position);
22
23
       cout << "After insertion:
                                     << result << endl;</pre>
```

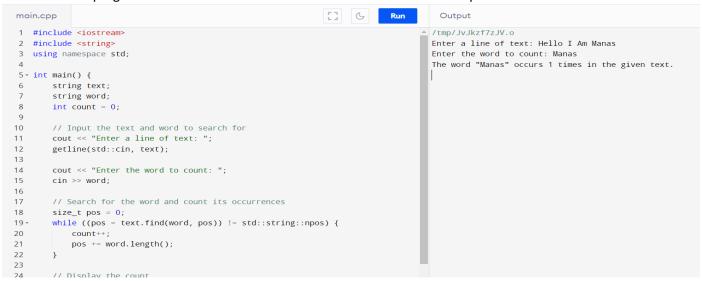
2# Write a C++ program to determine if the given string is a palindrome or not.

```
[] (
                                                                              Run
                                                                                         Output
 main.cpp
                                                                                      ▲ /tmp/JvJkzf7zJV.o
 3 #include <algorithm>
                                                                                        Enter a string: Hello
 4 using namespace std;
                                                                                        Hello is not a palindrome.
 6 // Function to check if a string is a palindrome
    bool is palindrome(string str)
 9
        string rev = str;
10
        reverse(rev.begin(), rev.end());
11
        return (str == rev);
12 }
13
14 int main()
        string str;
        cout << "Enter a string: ";
cin >> str;
17
18
19
20 -
        if (is_palindrome(str)) {
            cout << str << " is a palindrome." << endl;</pre>
        } else {
23
            cout << str << " is not a palindrome." << endl;</pre>
24
       }
25
26 return 0:
```

3# Write a C++ program to find a string within a sentence and replace it with another string.

```
[] G Run
1 #include <iostream>
                                                                                            /tmp/JvJkzf7zJV.o
                                                                                            Demonstration of replace
2 using namespace std;
                                                                                            Hello World !
4 // Function for demonstration
                                                                                            Hello Geeks !
5 void replaceDemo(string s1, string s2, string s3, string s4)
                                                                                            Hello to all !
                                                                                            Hello to all!!!!
6 - {
        // Replaces 7 characters from 0th index by s2
        s1.replace(0, 7, s2);
9
        cout << s1 << endl;
10
        // Replaces 3 characters from 0th index with "Hello"
        s4.replace(0, 3, "Hello ");
cout << s4 << endl;
13
14
        // Replaces 5 characters from 6th index of s4 with
        // 5 characters from 0th of s3
17
        s4.replace(6, 5, s3, 0, 5);
18
        cout << s4 << endl:
19
        // Replaces 5 characters from 6th index of s4 with
        // 6 characters from string "to all"
s4.replace(6, 5, "to all", 6);
22
        cout << s4 << endl;
23
        // Replaces 1 character from 12th index of s/ with
```

4# Write a C++ program that reads a line of text and counts all occurrence of a particular word.



5# Write a C++ program that displays the position or index in the string S where the string T begins, or 1 if S doesn't contain T.

```
Run
main.cpp
                                                                        G
                                                                                          Output
 1 #include <iostream>
                                                                                         /tmp/JvJkzf7zJV.o
   #include <string>
                                                                                         Enter the string S: ManasMaheshwari
 3 using namespace std;
                                                                                         Enter the string T: ana
                                                                                         String T begins at position 2 in string S.
 5 * int main() {
       string S, T;
cout << "Enter the string S: ";
 6
 8
        cin >> S:
        cout << "Enter the string T: ";</pre>
 9
10
       cin >> T;
11
12
        size_t found = S.find(T); // Search for T in S
13
        if (found != std::string::npos) {
15
            // If T is found in S, display the position (index) where it begins
            cout << "String T begins at position " << found + 1 << " in string S." <<</pre>
16
               endl:
17 -
18
            // If T is not found in S. display 1
            cout << "String T is not found in string S, so the result is 1." << endl;
19
20
21
22
        return 0;
23
```

1# Write C programs that use both recursive and non-recursive functions to find:

a) The factorial of a given integer.

```
Output
 main.cpp
                                                                      [] G Run
 1 #include <stdio.h>
                                                                                                /tmp/JvJkzf7zJV.o
                                                                                               Enter a non-negative integer: 23 Factorial (Non-recursive): 862453760
 3 // Non-recursive function to find factorial
 4 - int factorialNonRecursive(int n) {
                                                                                               Factorial (Recursive): 862453760
         int result = 1;
for (int i = 1; i <= n; i++) {
    result *= i;</pre>
         return result;
10 }
11
   // Recursive function to find factorial
13 - int factorialRecursive(int n) {
14* if (n == 0 | | n == 1) {
             return 1;
16 -
       } else {
            return n * factorialRecursive(n - 1);
17
19 }
20
         int num;
23 printf("Enter a non-negative integer: ");
```

b) To find the greatest common divisor of two given integers.

```
[] G Run
1 #include <stdio.h>
                                                                                      /tmp/JvJkzf7zJV.o
                                                                                      Enter two positive integers: 23
3 // Non-recursive function to find GCD
4 - int gcdNonRecursive(int a, int b) {
                                                                                      GCD (Non-recursive): 1
       int temp;
                                                                                      GCD (Recursive): 1
       while (b != 0) {
         temp = b;
           b = a % b;
a = temp;
8
       return a;
12 }
13
14 // Recursive function to find GCD
15 - int gcdRecursive(int a, int b) {
16 -
       if (b == 0) {
           return a;
17
18 -
       } else {
          return gcdRecursive(b, a % b);
20
21 }
22
23 - int main() {
```

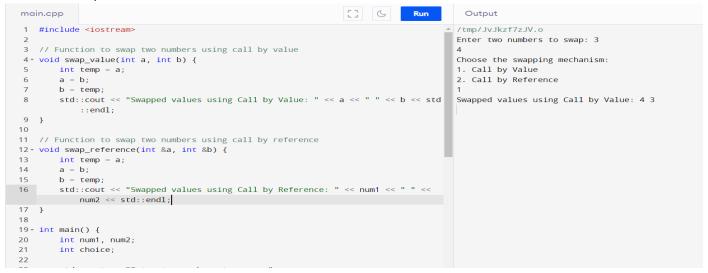
2# Write C programs that use both recursive and non-recursive functions to solve towers of Hanoi problem.

```
Output
main.cpp
                                                                             Run
 1 #include <stdio.h>
                                                                                      /tmp/JvJkzf7zJV.o
                                                                                      Enter the number of disks: 3
                                                                                      Move disk 1 from A to C
 3 - void towersOfHanoi(int n, char source, char auxiliary, char target) {
      if (n == 1) {
                                                                                      Move disk 2 from A to B
 5
            printf("Move disk 1 from %c to %c\n", source, target);
                                                                                      Move disk 1 from C to B
                                                                                      Move disk 3 from A to C
            return;
 6
 7
                                                                                      Move disk 1 from B to A
                                                                                      Move disk 2 from B to C
 8
       towersOfHanoi(n - 1, source, target, auxiliary);
                                                                                      Move disk 1 from A to C
10
        printf("Move disk %d from %c to %c\n", n, source, target);
        towersOfHanoi(n - 1, auxiliary, source, target);
11
12 }
13
14 * int main() {
15
      int numDisks;
        printf("Enter the number of disks: ");
16
17
       scanf("%d", &numDisks);
18
19 -
      if (numDisks < 1) {</pre>
20
           printf("Number of disks should be at least 1.\n");
21 -
       } else {
22
            towersOfHanoi(numDisks, 'A', 'B', 'C');
23
24
25
    return 0:
```

3# Write a C++ program to print the transpose of a given matrix using function.

```
[] (
                                                                                                                                                   Output
                                                                                                                                              ▲ /tmp/JvJkzf7zJV.o
             int matrix[100][100];
                                                                                                                                                 Enter the number of rows: 3
Enter the number of columns: 3
Enter the elements of the matrix:
             int rows, cols;
             std::cout << "Enter the number of rows: ";
std::cin >> rows;
std::cout << "Enter the number of columns: ";</pre>
             std::cin >> cols;
             if (rows <= 0 || cols <= 0) {
    std::cout << "Invalid matrix dimensions. Please enter positive values</pre>
33 -
                    for rows and columns.\n"; return 1;
36
             std::cout << "Enter the elements of the matrix:\n";
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < cols; j++) {
        std::cin >> matrix[i][j];
}
38
40 -
                                                                                                                                                 4 9 5
42
43
44
45
             transposeMatrix(matrix, rows, cols);
46
47
            return 0;
```

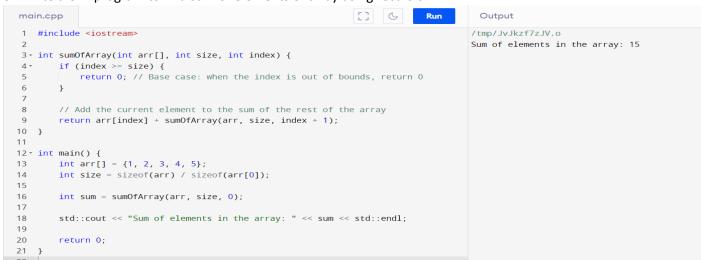
4# Write a C++ program to swap two number by both call by value and call by reference mechanism, using two functions swap_value() and swap_reference respectively, by getting the choice from the user and executing the user's choice by switch-case.



5# Write a C++ program to display all array elements using recursion.

```
[] 6
                                                                             Run
                                                                                        Output
   #include <iostream>
                                                                                       /tmp/JvJkzf7zJV.o
                                                                                      Array elements using recursion: 1 2 3 4 5
 3 - void displayArray(int arr[], int size, int index) {
4 -
       if (index >= size) {
5
            return; // Base case: stop when the index is out of bounds
 6
        // Print the element at the current index
       std::cout << arr[index] << " ";
10
        // Recursively call the function for the next index
11
12
       displayArray(arr, size, index + 1);
13 }
14
15 - int main() {
      int arr[] = {1, 2, 3, 4, 5};
16
17
        int size = sizeof(arr) / sizeof(arr[0]);
18
        std::cout << "Array elements using recursion: ";</pre>
19
20
        displayArray(arr, size, 0); // Start from index 0
21
22
        return 0;
```

6# Write a C++ program to find sum of elements of array using recursion.



7# Write a C++ program to find maximum and minimum elements in array using recursion.

```
Run
                                                                                    Output
1 #include <iostream>
                                                                                   /tmp/JvJkzf7zJV.o
                                                                                   Maximum element in the array: 9
3 // Function to find the maximum element in an array using recursion
                                                                                   Minimum element in the array: 1
4 - int findMax(int arr[], int size) {
       if (size == 1) {
          return arr[0]; // Base case: if there's only one element, it's the
6
              maximum
     int maxRest = findMax(arr, size - 1); // Find the maximum in the rest of
9
          the array
10
11
       return (maxRest > arr[size - 1]) ? maxRest : arr[size - 1];
12 }
14 // Function to find the minimum element in an array using recursion
15 - int findMin(int arr[], int size) {
     if (size == 1) {
16 -
          return arr[0]; // Base case: if there's only one element, it's the
              minimum
18
19
       int minRest = findMin(arr, size - 1); // Find the minimum in the rest of
```

1# Write a C++ program that uses functions to perform the following operations:

- Reading a complex number
- ii. ii. Writing a complex number
- iii. Addition and subtraction of two complex numbers iv. Multiplication of two complex numbers. Note: represent complex number using a structure.

```
[] G Run
 1 #include <iostream>
                                                                                                              /tmp/JvJkzf7zJV.o
                                                                                                             Enter details for the first complex number: Enter real part: \ensuremath{\mathsf{2}}
   using namespace std;
                                                                                                             Enter imaginary part: 3
Enter details for the second complex number:
     // Structure to represent a complex number
    struct Complex {
                                                                                                             Enter real part:
         double real;
double imag;
                                                                                                             Enter imaginary part: 7
                                                                                                             Sum of the two complex numbers:
Complex Number: 7 + 10i
    // Function to read a complex number
12 void readComplexNumber(Complex &num) {
13     cout << "Enter real part: ";</pre>
                                                                                                             Difference of the two complex numbers:
Complex Number: -3 + -4i
         cout << "Enter re
cin >> num.real;
          cout << "Enter imaginary part: ";</pre>
                                                                                                             Product of the two complex numbers:
         cin >> num.imag;
                                                                                                             Complex Number: -11 + 29i
16
   }
    // Function to write a complex number
    void writeComplexNumber(const Complex &num) {
   cout << "Complex Number: " << num.real << " + " << num.imag << "i" << endl;</pre>
    // Function to add two complex numbers
```

2# Write a C++ program to compute the monthly pay of 100 employees using each employee's name, basic pay. The DA is computed as 52% of the basic pay. Gross-salary (basic pay + DA). Print the employees name and gross salary

```
main.cpp
                                                                                   Run
                                                                                                 Output
                                                                                              ▲ /tmp/JvJkzf7zJV.o
13 - double computeDA(double basicPay) {
                                                                                                Enter details for Employee 1:
        return 0.52 * basicPay;
                                                                                                Name: Manas
                                                                                                Basic Pay: 90000
                                                                                                Enter details for Employee 2:
   // Function to compute the gross salary
                                                                                                Name: Anamika
18 double computeGrossSalary(double basicPay, double da) {
                                                                                                Basic Pay: 55000
        return basicPay + da;
                                                                                                Enter details for Employee 3:
20 }
                                                                                                Name: Arushi
                                                                                                Basic Pay: 60000
22 // Function to print employee details and gross salary
23 - void printEmployeeDetails(const Employee & Employee, double grossSalary) {
24     cout < "Employee Name: " < employee.name << "\tGross Salary: " <<
                                                                                                Employee Name: Manas
                                                                                                                          Gross Salary: 136800
                                                                                                Employee Name: Anamika Gross Salary: 83600
                                                                                                Employee Name: Arushi Gross Salary: 91200
             grossSalary << endl;</pre>
25 }
26
27 * int main() {
         const int numEmployees = 3;
28
         Employee employees[numEmployees];
30
     // Input basic pay and name for each employee
        for (int i = 0; i < numEmployees; ++i) {
   cout << "Enter details for Employee " << i + 1 << ":\n";
             getline(cin >> ws, employees[i].name); // Use getline to allow spaces
```

3# Create a Book structure containing book_id, title, author name and price. Write a C++ program to pass a structure as a function argument and print the book details.

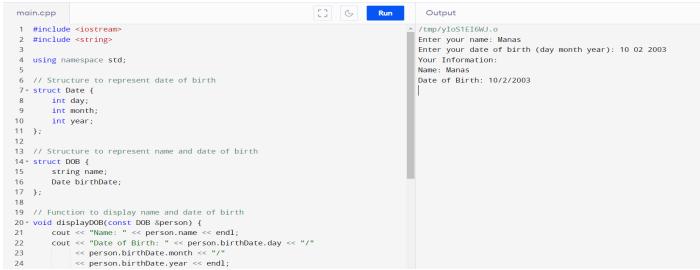
```
main.cpp
                                                                    Run
                                                                                             Output

↑ tmp/9z3nIl5i5k.o
22 * int main() {
         // Create a Book structure
                                                                                            Enter Book ID: 1
                                                                                            Enter Title: Harry Potter
24
       Book myBook;
                                                                                            Enter Author: J K Rowling
25
       // Input book details
cout << "Enter Book ID: ";</pre>
                                                                                            Enter Price: $29.9
26
                                                                                            Book Details:
27
                                                                                            Book ID: 1
28
       cin >> myBook.book_id;
                                                                                            Title: Harry Potter
29
                                                                                            Author: J K Rowling
       cout << "Enter Title: ";
30
                                                                                            Price: $29.9
       getline(cin >> ws, myBook.title); // Allowing spaces in the title using
            getline
32
        cout << "Enter Author: ";</pre>
34
       getline(cin >> ws, myBook.author); // Allowing spaces in the author's name
            using getline
35
        cout << "Enter Price: $";</pre>
36
37
        cin >> myBook.price;
38
        // Print book details using the function
40
        cout << "\nBook Details:\n";</pre>
        printBookDetails(myBook);
41
42
43
```

4# Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C++ program to display your present address



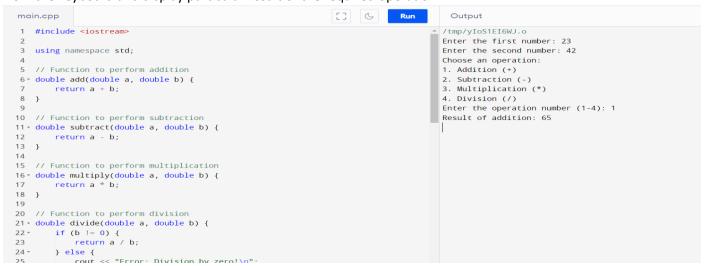
5# Write a C++ program to define a structure named D.O.B., which contains name, day, month and year. Using the concept of nested structures display your name and date of birth.



1# Write a program in C++ to display your name, Branch, Year on to the computer screen without using classes and object. All information should be displayed in the separate line.

```
[] 6
                                                                                        Run
                                                                                                       Output
main.cpp
    #include <iostream>
#include <string>
                                                                                                     /tmp/vIoS1FI6WJ.o
                                                                                                     Name: Manas
                                                                                                     Branch: CS
                                                                                                     Year: 2023
Name :Manas
 4 using namespace std;
 6 * int main() {
                                                                                                     Branch : CS
        string name, branch, year;
// Displaying name, branch, and year on separate lines
                                                                                                     Year :2023
         cout << "Name: ";
        cin >> name;
cout << "Branch: ";</pre>
10
11
        cin >> branch;
        cout << "Year: ";
13
14
        cin >>year;
        cout << "Name :"<<name<<"\n";
cout <<"Branch :"<<branch<"\n";</pre>
20
         return 0;
```

2# Write a menu driven program in C++ to perform all basic arithmetic operation addition, subtraction, multiplication, and division of two given values. Program receives two values and required operation to be performed from the keyboard and display particular result of the required operation.



3# Write a menu driven program in C++ that receives 4-digit integer value the keyboard and perform following operations:

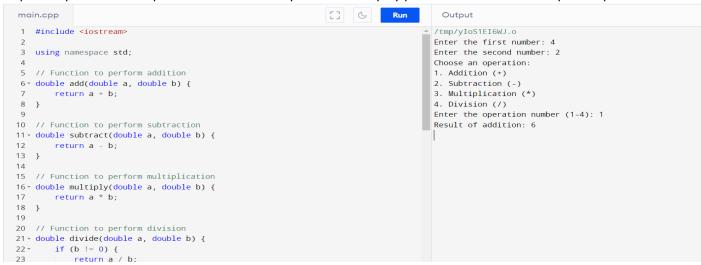
- i) Reverse of that no.
- ii) sum of number with its reverse.
- iii) sum of alternative digits (1 digit+3 digit and 2 digit+4 digit)

```
1 #include <iostream>
                                                                                         /tmp/vToS1FT6WJ.o
                                                                                        Enter a 4-digit integer: 1234
                                                                                        Choose an operation:
 3 using namespace std;
                                                                                        i) Reverse of the number
 5 // Function to reverse a 4-digit number
                                                                                        ii) Sum of number with its reverse
                                                                                        iii) Sum of alternative digits
 6 - int reverseNumber(int num) {
       int reversedNum = 0;
                                                                                        Enter your choice (i/ii/iii): i
       while (num > 0) {
                                                                                        Reverse of 1234 is: 4321
           reversedNum = reversedNum * 10 + num % 10:
 9
10
            num /= 10;
12
        return reversedNum;
13 }
14
16 - int sumOfDigits(int num) {
       int sum = 0;
while (num > 0) {
17
18 -
        sum += num % 10;
            num /= 10;
```

4# Write a menu driven program in C++ to receive integer number and convert equivalent binary, octal, hexadecimal number.

```
[] G Run
main.cpp
                                                                                          Output
                                                                                         /tmp/yIoS1EI6WJ.o
    #include <iostream>
   #include <iomanip>
                                                                                         Enter an integer number: 4
                                                                                         Choose a base for conversion:
4 using namespace std;
                                                                                         1. Binary
                                                                                         2. Octal
   // Function to convert decimal to binary
                                                                                         3. Hexadecimal
7 - string decimalToBinary(int decimal) {
                                                                                         Enter your choice (1-3): 1
       string binary =
                                                                                         Binary representation: 100
       while (decimal > 0)
          binary = char('0' + decimal % 2) + binary;
decimal /= 2;
10
11
       return binary.empty() ? "0" : binary;
14 }
15
16 // Function to convert decimal to octal
17 - string decimalToOctal(int decimal) {
       string octal = "";
while (decimal > 0) {
18
       octal = char('0' + decimal % 8) + octal;
            decimal /= 8;
        return octal.emntv() ? "O" · octal·
```

5# Write a menu driven program in C++ to perform all basic arithmetic operation addition, subtraction, multiplication, and division of two given values using function and switch case. Program receives two values and required operation to be performed from the keyboard and display particular result of the required operation.



6# Define a class Bank Account to represent a bank account. Include the following members: Data Members:

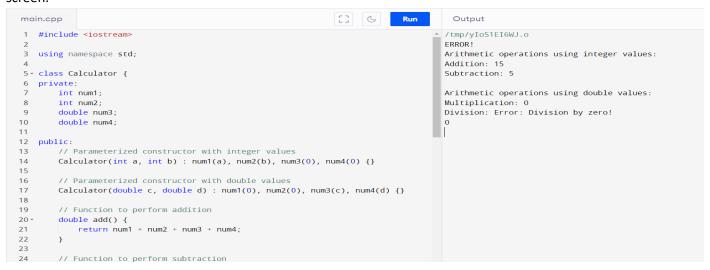
o Name of the depositor o Account Number o Type of account o Balance amount in the account Member Functions: o To assign initial value o To deposit an amount o To withdraw an amount after checking

```
Run
 1 #include <iostream>
                                                                                        /tmp/yIoS1EI6WJ.o
 2 #include <string>
                                                                                        Deposit successful. Updated balance: $1500.5
                                                                                        Withdrawal successful. Updated balance: $1299.75
 4 using namespace std;
                                                                                        Account Information:
                                                                                        Depositor Name: John Doe
 6 - class BankAccount {
                                                                                        Account Number: 123456789
                                                                                        Account Type: S
8
        string depositorName;
                                                                                        Current Balance: $1299.75
       long long accountNumber;
10
       char accountType;
11
       double balance;
12
13 public:
14
       // Constructor to initialize data members
15 -
        BankAccount (const\ string \&\ name,\ long\ long\ acc Number,\ char\ acc Type,\ double
            initialBalance) {
            depositorName = name:
            accountNumber = accNumber;
17
18
            accountType = accType;
19
            balance = initialBalance;
20
21
22
        // Function to deposit an amount
23 -
       void deposit(double amount) {
```

1# Write a program in C++ to demonstrate default constructor. Create a class having two data members in the private section. Define a default constructor to initialize these data members to initial value and display these values with the help of member function

```
Run
                                                                 [] 6
                                                                                         Output
 1 #include <iostream>
                                                                                        /tmp/yToS1EI6WL.o
                                                                                        Initialized values using the default constructor:
                                                                                        Value of num1: 10
 3 using namespace std;
                                                                                        Value of num2: 5.5
 5 - class MvClass {
       int num1:
       double num2:
10 public:
       // Default constructor
       MyClass() {
13
            // Initialize data members to initial values
           num1 = 10;
num2 = 5.5;
14
16
       // Member function to display data members
19 -
       void displayValues() {
                    "Value of num1: " << num1 << end1;
20
           court <<
            cout << "Value of num2: " << num2 << end1;
22
23
   };
```

2# Write a program in C++ to demonstrate parameterized/constructor overloading constructor. Create a class calculator that contains four data members in it. Initialize data members with different values using parameterized constructor and perform various arithmetic operation over these values and display result on to the computer screen.



3# Create a class called Triangle that stores the length of the base and height of a right triangle in two private instance variables. Include a constructor that sets these values. Define two functions. The first is hypo(), which returns the length of the hypotenuse. The second is area (), which returns the area of the triangle

```
[] (
                                                                                    Run
                                                                                                 /tmp/yIoS1EI6WJ.o
   #include <iostream>
    #include <cmath>
                                                                                                 Hypotenuse length: 5
                                                                                                 Area of the triangle: 6
 4 using namespace std;
6 - class Triangle {
        double base:
10
        // Constructor to set the base and height
Triangle(double baseValue, double heightValue) : base(baseValue), height
13
             (heightValue) {}
14
         // Function to calculate and return the length of the hypotenuse
        double hypo() const {
16 -
            return sqrt(base * base + height * height);
18
19
20
        // Function to calculate and return the area of the triangle
        double area() const {
   return 0.5 * base * height;
22
```

4# Create a class for counting the number of objects created and destroyed within various block using constructor and destructors.

```
[] 6
main.cpp
                                                                           Run
                                                                                      Output
 1 #include <iostream>
                                                                                     /tmp/yIoS1EI6WJ.o
                                                                                     Object 1 created.
 3 using namespace std;
                                                                                     Object 2 created.
                                                                                     Number of objects in block 1: 2
 5 - class ObjectCounter {
                                                                                    Object 2 destroyed.
                                                                                    Object 1 destroyed.
 6 private:
       static int objectCount; // Static variable to count the number of objects
                                                                                    Object 3 created.
        int objectId; // Unique ID for each object
                                                                                    Object 4 created.
 9
                                                                                     Object 5 created.
                                                                                    Number of objects in block 2: 5
10 public:
11 -
        ObjectCounter() {
                                                                                    Object 5 destroyed.
12
           objectId = ++objectCount; // Increment object count and assign ID
                                                                                     Object 4 destroyed.
            cout << "Object " << objectId << " created." << endl;</pre>
13
                                                                                    Object 3 destroyed.
14
                                                                                    Number of objects outside any block: 5
15
16 -
     ~ObjectCounter() {
            cout << "Object " << objectId << " destroyed." << endl;
17
18
19
20 -
      static int getObjectCount() {
21
           return objectCount;
22
23 };
24
25 // Initialize the static variable
```