**Easy 13 date**

**1-Write a C++ program to declare and initialize a constant integer with value 10.**

#include <iostream>

using namespace std;

int main()

{

int n;

cout<<"enter the const number";

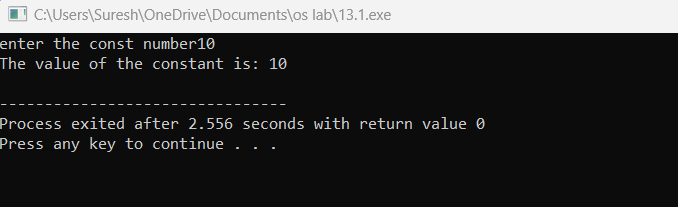
cin>>n;

cout << "The value of the constant is: " <<n <<endl;

return 0;

}

**Output:**

****

**2-Write a C++ program to demonstrate the use of integer, float, and character data types**

#include <iostream>

using namespace std;

int main() {

int n;

cout << "Integer value: ";

cin>>n;

float m ;

cout << "Float value: ";

cin>>m;

char c;

cout << "Character value: ";

cin>>c;

cout<<"integer value is :"<<n<<endl;

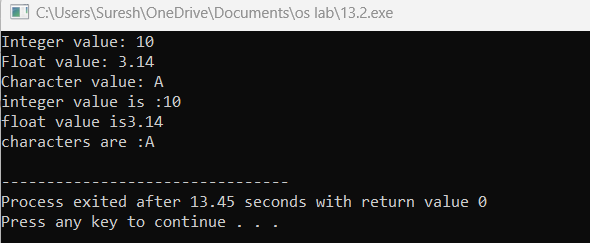
cout<<"float value is"<<m<<endl;

cout<<"characters are :"<<c<<endl;

return 0;

}

**Output:**

****

**3**. **Create a C++ program to declare an integer variable and a float variable, then assign the integer variable to the float variable**

#include <iostream>

using namespace std;

int main() {

int n;

float m;

cout<<"enter the integer value";

cin>>n;

cout<<"enter the float value";

cin>>m;

m = static\_cast<float>(n);

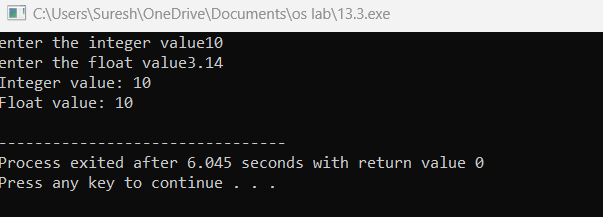
cout << "Integer value: " << n <<endl;

cout << "Float value: " << m<<endl;

return 0;

}

**Output:**

****

**4-Implement a C++ program that performs an implicit conversion from integer to float**

#include<iostream>

using namespace std;

int main()

{

int n;

float m;

cout<<"enter the integer";

cin>>n;

cout<<"enter float value";

cin>>m;

m=n;

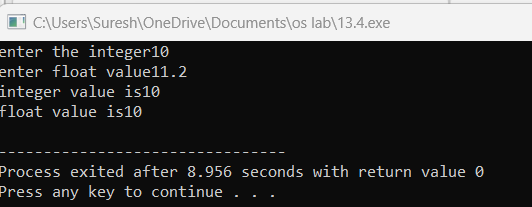
cout<<"integer value is"<<n<<endl;

cout<<"float value **is"<<m<<endl;**

**return 0;**

**}**

**Output;**

****

**5-Write a C++ program to declare and initialize two integer variables and perform addition using the + operator**

#include<iostream>

using namespace std;

int main()

{

int num1,num2,sum;

cout<<"enter the num1";

cin>>num1;

cout<<"enter the num2";

cin>>num2;

sum=num1+num2;

cout<<"num1"<<num1<<endl;

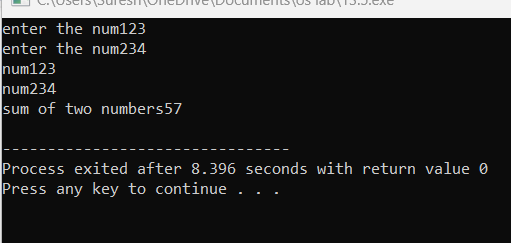
cout<<"num2"<<num2<<endl;

cout<<"sum of two numbers"<<sum<<endl;

return 0;

**}**

**Output;**

****

6-**Create a C++ program to demonstrate the use of if-else control structure to determine if a number is even or odd**

#include <iostream>

using namespace std;

int main() {

int number;

cout << "Enter a number: ";

cin >> number;

if (number % 2 == 0)

{

cout << number << " is even." << endl;

}

else

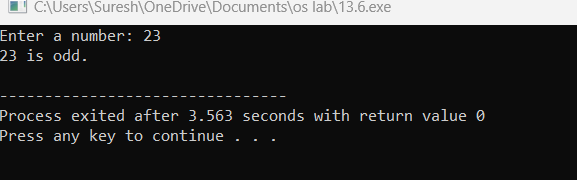
{

cout << number << " is odd." << endl;

}

return 0;

}



7-Write a C++ program to declare an array of integers and display its elements using a loop.

#include <iostream>

using namespace std;

int main() {

int a[] = {25, 50, 75, 100};

cout << a[0] << '\n';

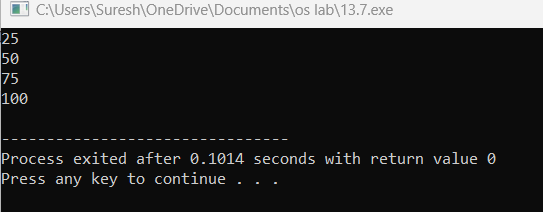
cout << a[1] << '\n';

cout << a[2] << '\n';

cout << a[3] << '\n';

return 0;

}



8-**Implement a C++ program to demonstrate the use of switch-case control structure to display the name of the day based on the user input (1-7).**

#include<iostream>

using namespace std;

int main()

{

int day;

cout<<"\nEnter the Day's number :";

cin>>day;

switch (day)

{

case 1:

cout<<"Monday";

break;

case 2:

cout<<"Tuesday";

break;

case 3:

cout<<"Wednesday";

break;

case 4:

cout<<"Thursday";

break;

case 5:

cout<<"Friday";

break;

case 6:

cout<<"Saturday";

break;

case 7:

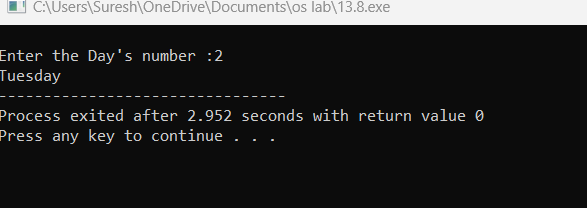
cout<<"Sunday";

break;

}

return 0;

}



9-Create a C++ program to calculate the area of a rectangle using user input for length and width

#include <iostream>

using namespace std;

int main() {

double length, width;

cout << "Enter the length of the rectangle: ";

cin >> length;

cout << "Enter the width of the rectangle: ";

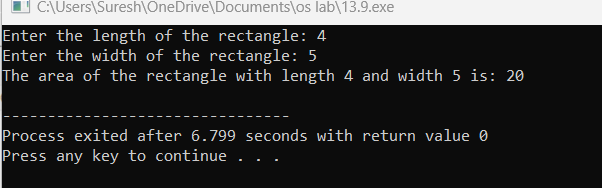
cin >> width;

double area = length \* width;

cout << "The area of the rectangle with length " << length << " and width " << width << " is: " << area <<endl;

return 0;

}



10-Write a C++ program to find the maximum of three numbers using conditional statements.

#include <iostream>

using namespace std;

int main() {

int num1, num2, num3;

cout << "Enter three numbers: ";

cin >> num1 >> num2 >> num3;

int max = num1;

if (num2 > max) {

max = num2;

}

if (num3 > max) {

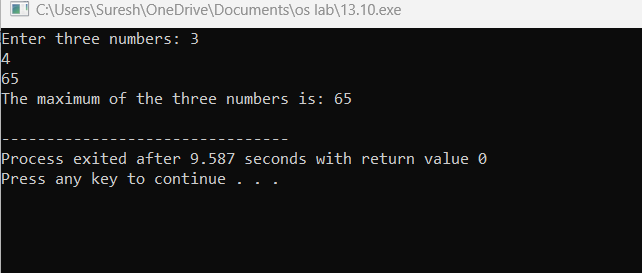
max = num3;

}

std::cout << "The maximum of the three numbers is: " << max << std::endl;

return 0;

}



11-Write a C++ program that calculates the area of a circle. Use a constant variable for the value of pi and prompt the user to enter the radius. Display the calculated area.

#include <iostream>

using namespace std;

int main() {

float radius, area;

cout << "Enter the radius of circle: ";

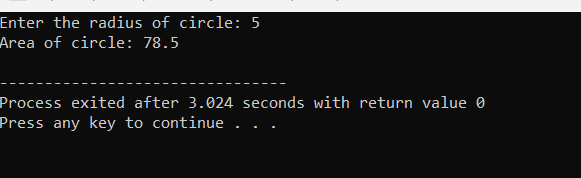
cin >> radius;

area = 3.14 \* radius \* radius;

cout << "Area of circle: " << area<< endl;

return 0;

}



12-Develop a C++ program that reads two numbers from the user, one integer and one float. Perform division of the float by the integer and display the result. Ensure proper type compatibility and implicit conversion handling.

#include <iostream>

int main() {

int integerNumber;

float floatNumber;

std::cout << "Enter an integer: ";

std::cin >> integerNumber;

std::cout << "Enter a float: ";

std::cin >> floatNumber;

if (integerNumber != 0) {

float result = floatNumber / integerNumber;

std::cout << "Result of float divided by integer: " << result << std::endl;

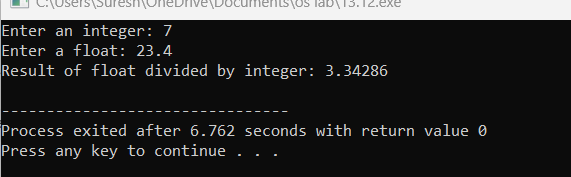
} else {

std::cout << "Error: Division by zero is not allowed." << std::endl;

}

return 0;

}



13-Write a C++ program to determine whether a given year is a leap year or not. Use logical operators and appropriate control structures to implement the leap year logic.

#include <iostream>

using namespace std;

int main() {

int year;

cout << "Enter a year: ";

cin >> year;

if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {

cout << year << " is a leap year.";

}

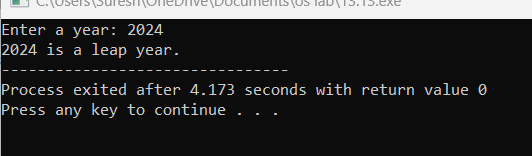
else {

cout << year << " is not a leap year.";

}

return 0;

}



14-Create a C++ program that calculates the area of a rectangle. Prompt the user to enter the length and width of the rectangle as floating-point numbers, and display the calculated area.

#include <iostream>

using namespace std;

int main() {

float length, width;

cout << "Enter the length of the rectangle: ";

cin >> length;

cout << "Enter the width of the rectangle: ";

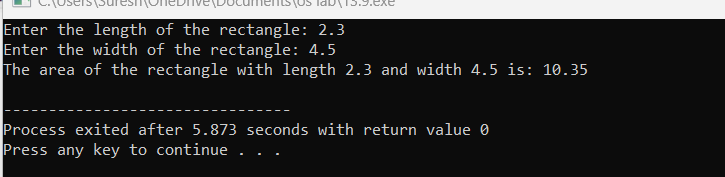
cin >> width;

float area = length \* width;

cout << "The area of the rectangle with length " << length << " and width " << width << " is: " << area <<endl;

return 0;

}



15-

Develop a C++ program that reads an integer from the user and checks if it is an odd number. Use bitwise AND operator and handle implicit conversion properly.

#include <iostream>

bool isOdd(int num) {

return num & 1;

}

int main() {

int num;

std::cout << "Enter an integer: ";

std::cin >> num;

if (isOdd(num))

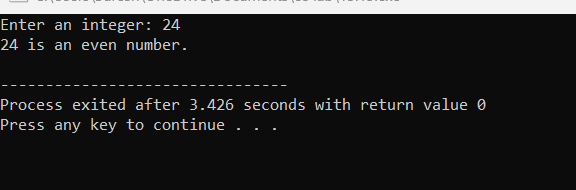
std::cout << num << " is an odd number." << std::endl;

else

std::cout << num << " is an even number." << std::endl;

return 0;

}



16-Write a C++ program that prompts the user to enter a month number (1-12) and displays the corresponding month name. Use a switch-case statement to implement this.

#include <iostream>

int main() {

int monthNumber;

// Prompt the user to enter a month number

std::cout << "Enter a month number (1-12): ";

std::cin >> monthNumber;

// Check if the entered month number is within the valid range (1-12)

if (monthNumber >= 1 && monthNumber <= 12) {

// Use a switch-case statement to display the corresponding month name

switch (monthNumber) {

case 1:

std::cout << "January" << std::endl;

break;

case 2:

std::cout << "February" << std::endl;

break;

case 3:

std::cout << "March" << std::endl;

break;

case 4:

std::cout << "April" << std::endl;

break;

case 5:

std::cout << "May" << std::endl;

break;

case 6:

std::cout << "June" << std::endl;

break;

case 7:

std::cout << "July" << std::endl;

break;

case 8:

std::cout << "August" << std::endl;

break;

case 9:

std::cout << "September" << std::endl;

break;

case 10:

std::cout << "October" << std::endl;

break;

case 11:

std::cout << "November" << std::endl;

break;

case 12:

std::cout << "December" << std::endl;

break;

}

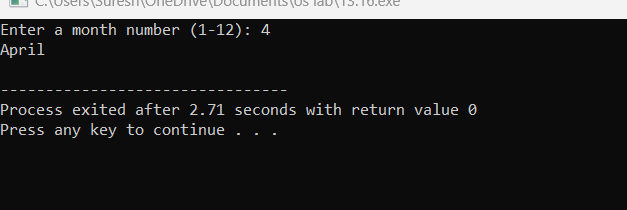
} else {

std::cout << "Invalid month number. Please enter a number between 1 and 12." << std::endl;

}

return 0;

}



17-Create a C++ program that calculates the volume of a sphere. Prompt the user to enter the radius and use a constant variable for the value of pi. Display the calculated volume.

#include <iostream>

using namespace std;

int main() {

const double PI = 3.14159;

double radius;

cout << "Enter the radius of the sphere: ";

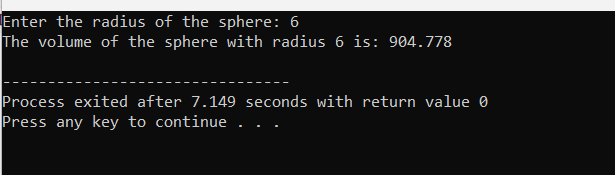
cin >> radius;

double volume = (4.0 / 3.0) \* PI \* radius \* radius \* radius;

cout << "The volume of the sphere with radius " << radius << " is: " << volume << endl;

return 0;

}



18-Develop a C++ program that reads two integers from the user and performs division. Ensure that if the second number is zero, the program should not crash, but display an appropriate error message.

#include <iostream>

using namespace std;

int main() {

int dividend, divisor;

cout << "Enter the dividend: ";

cin >> dividend;

cout << "Enter the divisor: ";

cin >> divisor;

if (divisor == 0) {

cout << "Error: Division by zero is not allowed." <<endl;

} else {

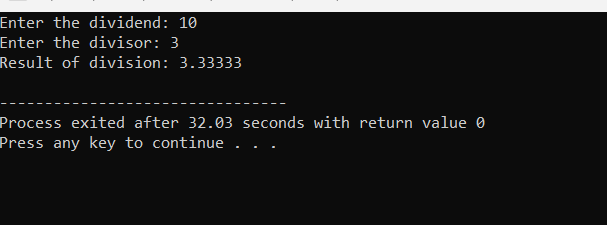
double quotient = static\_cast<double>(dividend) / divisor;

cout << "Result of division: " << quotient <<endl;

}

return 0;

}



19-Write a C++ program that defines a class representing a complex number. Implement operator overloading for addition and subtraction of complex numbers.

#include <iostream>

class Complex {

private:

double real;

double imaginary;

public:

Complex(double realPart = 0.0, double imaginaryPart = 0.0) : real(realPart), imaginary(imaginaryPart) {}

Complex operator+(const Complex& other) const {

return Complex(real + other.real, imaginary + other.imaginary);

}

Complex operator-(const Complex& other) const {

return Complex(real - other.real, imaginary - other.imaginary);

}

void display() const {

std::cout << real << " + " << imaginary << "i";

}

};

int main() {

Complex num1(3.0, 4.0);

Complex num2(1.5, 2.5);

Complex sum = num1 + num2;

std::cout << "Sum: ";

sum.display();

std::cout << std::endl;

Complex difference = num1 - num2;

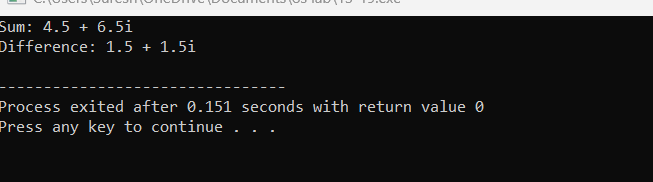
std::cout << "Difference: ";

difference.display();

std::cout << std::endl;

return 0;

}



20-. Create a C++ program that prompts the user to enter a number and calculates the factorial of that number using a loop. Display the result.

#include <iostream>

int main() {

int number;

unsigned long long factorial = 1;

std::cout << "Enter a number: ";

std::cin >> number;

if (number < 0) {

std::cout << "Error: Factorial is not defined for negative numbers." << std::endl;

} else {

for (int i = 1; i <= number; ++i) {

factorial \*= i;

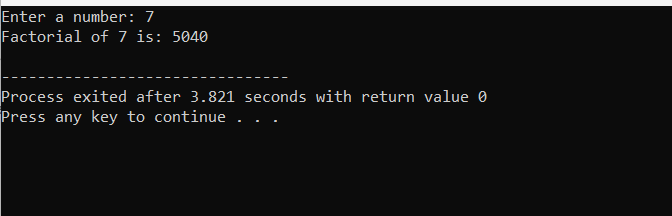
}

std::cout << "Factorial of " << number << " is: " << factorial << std::endl;

}

return 0;

}



21-Write a C++ program to implement a recursive function to calculate the Fibonacci sequence up to a given number 'n'.

#include <iostream>

using namespace std;

int fib(int n) {

if((n==1)||(n==0)) {

return(n);

}else {

return(fib(n-1)+fib(n-2));

}

}

int main() {

int n , i=0;

cout << "Enter the number of terms of series : ";

cin >> n;

cout << "\nFibonnaci Series : ";

while(i < n) {

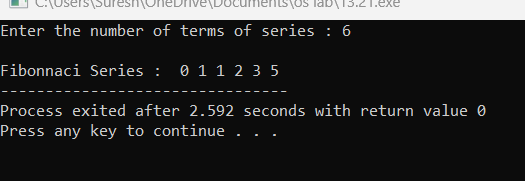
cout << " " << fib(i);

i++;

}

return 0;

}



22-

#include <iostream>

using namespace std;

int main ()

{

int\* m = NULL;

m = new(nothrow) int;

if (!m)

cout<< "allocation of memory failed\n";

else

{

\*m=29;

cout<< "Value of m: " << \*m <<endl;

}

float \*f = new float(75.25);

cout<< "Value of f: " << \*f <<endl;

int size = 5;

int \*arr = new(nothrow) int[size];

if (!arr)

cout<< "allocation of memory failed\n";

else

{

for (int i = 0; i< size; i++)

arr[i] = i+1;

cout<< "Value store in block of memory: ";

for (int i = 0; i< size; i++)

cout<<arr[i] << " ";

}

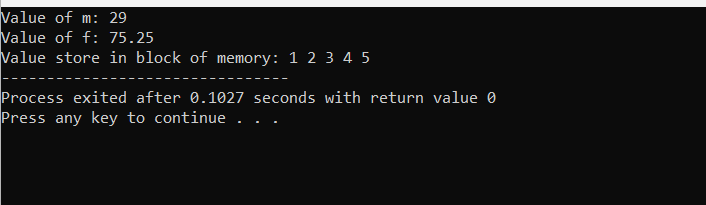
delete m;

delete f;

delete[] arr;

return 0;

}



24-Write a C++ program that demonstrates function overloading for different data types. Create overloaded functions to calculate the area of a rectangle, circle, and triangle.

#include <iostream>

#include <cmath>

using namespace std;

double calculateArea(double length, double width) {

return length \* width;

}

double calculateArea(double radius) {

return M\_PI \* radius \* radius;

}

float calculateArea(float base, float height)

{

return 0.5 \* base \* height;

}

int main() {

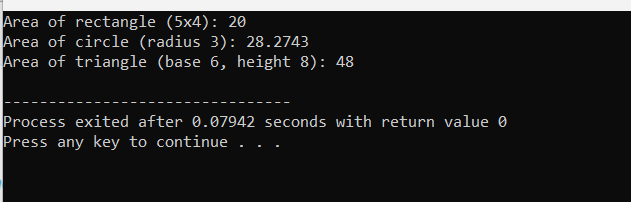
cout << "Area of rectangle (5x4): " << calculateArea(5.0, 4.0) << endl;

cout << "Area of circle (radius 3): " << calculateArea(3.0) << endl;

cout << "Area of triangle (base 6, height 8): " << calculateArea(6.0, 8.0) << endl;

return 0;

}



26-Write a C++ program that defines a template function to find the maximum element in an array of any data type. Test the function with integer, float, and double arrays

#include <iostream>

using namespace std;

template<typename T>

T findMax(const T arr[], int size) {

T maxElement = arr[0];

for (int i = 1; i < size; ++i) {

if (arr[i] > maxElement) {

maxElement = arr[i];

}

}

return maxElement;

}

int main() {

int intArray[] = {3, 7, 2, 9, 5};

int intMax = findMax(intArray, 5);

cout << "Maximum element in the integer array: " << intMax << endl;

float floatArray[] = {3.5f, 7.9f, 2.3f, 9.1f, 5.8f};

float floatMax = findMax(floatArray, 5);

cout << "Maximum element in the float array: " << floatMax << endl;

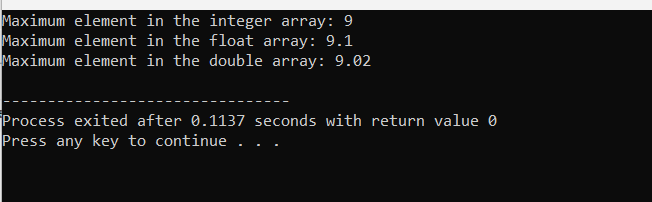
double doubleArray[] = {3.56, 7.81, 2.45, 9.02, 5.64};

double doubleMax = findMax(doubleArray, 5);

cout << "Maximum element in the double array: " << doubleMax << endl;

return 0;

}



27-Create a C++ program to implement a recursive function to solve the Tower of Hanoi puzzle with 'n' disks.

#include <iostream>

using namespace std;

void moveDisk(int disk, char source, char destination) {

cout << "Move disk " << disk << " from " << source << " to " << destination << endl;

}

void towerOfHanoi(int n, char source, char auxiliary, char destination) {

if (n == 1) {

moveDisk(n, source, destination);

return;

}

towerOfHanoi(n - 1, source, destination, auxiliary);

moveDisk(n, source, destination);

towerOfHanoi(n - 1, auxiliary, source, destination);

}

int main() {

int numDisks;

cout << "Enter the number of disks: ";

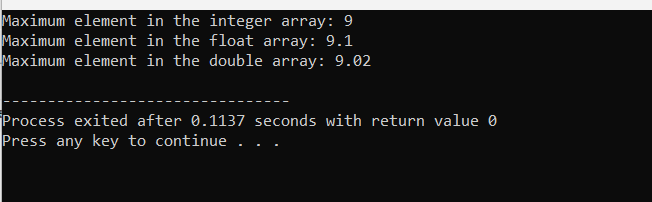
cin >> numDisks;

cout << "Steps to solve the Tower of Hanoi puzzle with " << numDisks << " disks:" << endl;

towerOfHanoi(numDisks, 'A', 'B', 'C');

return 0;

}



29-Write a C++ program that implements a callback mechanism using function pointers. Create a function that takes a callback function as an argument and calls it.

#include <iostream>

using namespace std;

typedef void (\*CallbackFunction)(int);

void performCallback(int value, CallbackFunction callback) {

cout << "Performing some operation..." << endl;

callback(value);

}

void callbackFunction1(int value) {

cout << "Callback function 1 called with value: " << value << endl;

}

void callbackFunction2(int value) {

cout << "Callback function 2 called with value: " << value << endl;

}

int main() {

int value = 42;

cout << "Calling performCallback with callbackFunction1:" << endl;

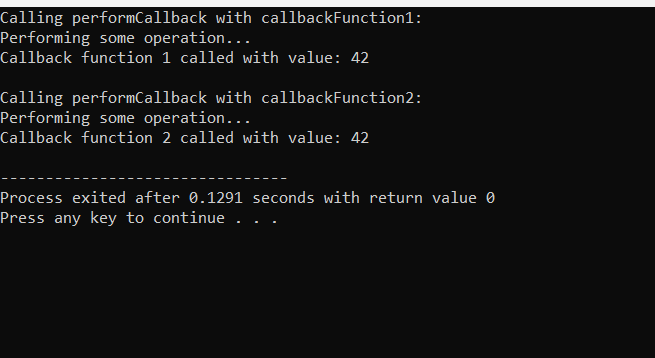
performCallback(value, callbackFunction1);

cout << "\nCalling performCallback with callbackFunction2:" << endl;

performCallback(value, callbackFunction2);

return 0;

}



30-. Develop a C++ program to implement a binary search tree (BST) and perform operations such as insertion, deletion, and traversal (in-order, pre-order, post-order).

#include <iostream>

using namespace std;

class Node {

public:

int data;

Node\* left;

Node\* right;

Node(int v)

{

this->data = v;

this->left = this->right = NULL;

}

};

void printInorder(Node\* node)

{

if (node == NULL)

return;

printInorder(node->left);

cout << node->data << " ";

printInorder(node->right);

}

int main()

{

Node\* root = new Node(100);

root->left = new Node(20);

root->right = new Node(200);

root->left->left = new Node(10);

root->left->right = new Node(30);

root->right->left = new Node(150);

root->right->right = new Node(300);

cout << "Inorder Traversal: ";

printInorder(root);

return 0;

}

