13/03/2024

DSA0187: OBJECT ORIENTED PROGRAMMING WITH C++

**N.Ravi Teja Reddy (192210667)**

1

#include<iostream>

using namespace std;

int main ()

{

const int value=10;

cout << value;

}

2

#include <iostream>

int main () {

int integerNum = 10;

std::cout << "Integer number: " << integerNum << std::endl;

float floatNum = 3.14;

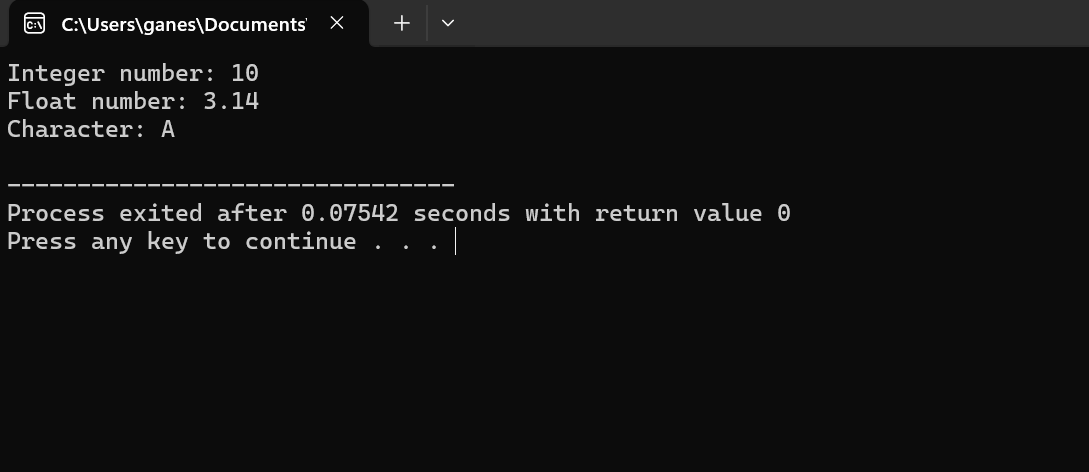
std::cout << "Float number: " << floatNum << std::endl;

char charValue = 'A';

std::cout << "Character: " << charValue << std::endl;

return 0;

}



3

#include <iostream>

int main () {

int integerVar = 10;

float floatVar;

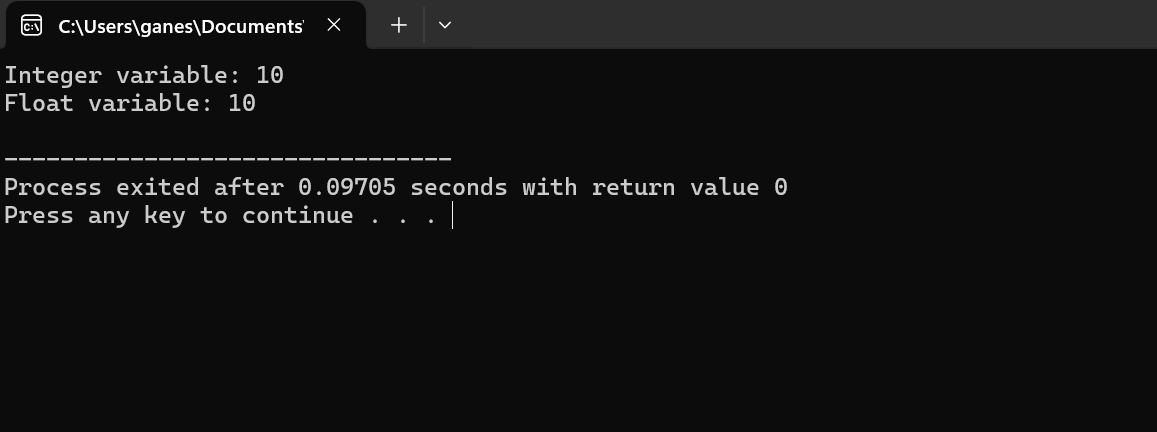
floatVar = integerVar;

std::cout << "Integer variable: " << integerVar << std::endl;

std::cout << "Float variable: " << floatVar ;

return 0;

}



4

#include <iostream>

int main() {

int integerValue = 10;

float floatValue;

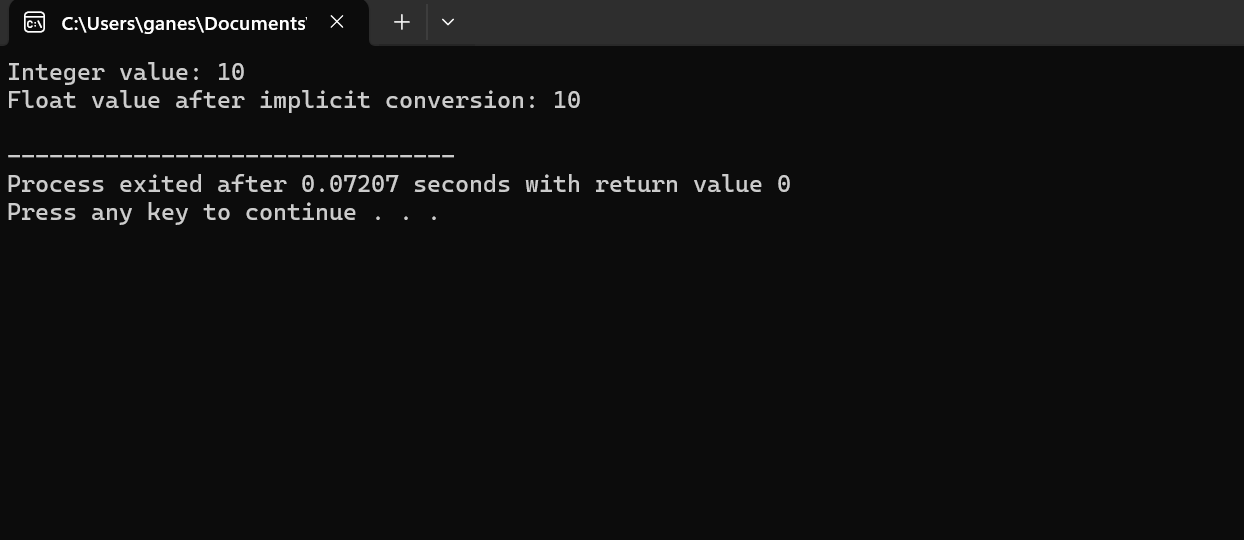
floatValue = integerValue;

std::cout << "Integer value: " << integerValue << std::endl;

std::cout << "Float value after implicit conversion: " << floatValue << std::endl;

return 0;

}



5.

#include<iostream>

using namespace std;

int main()

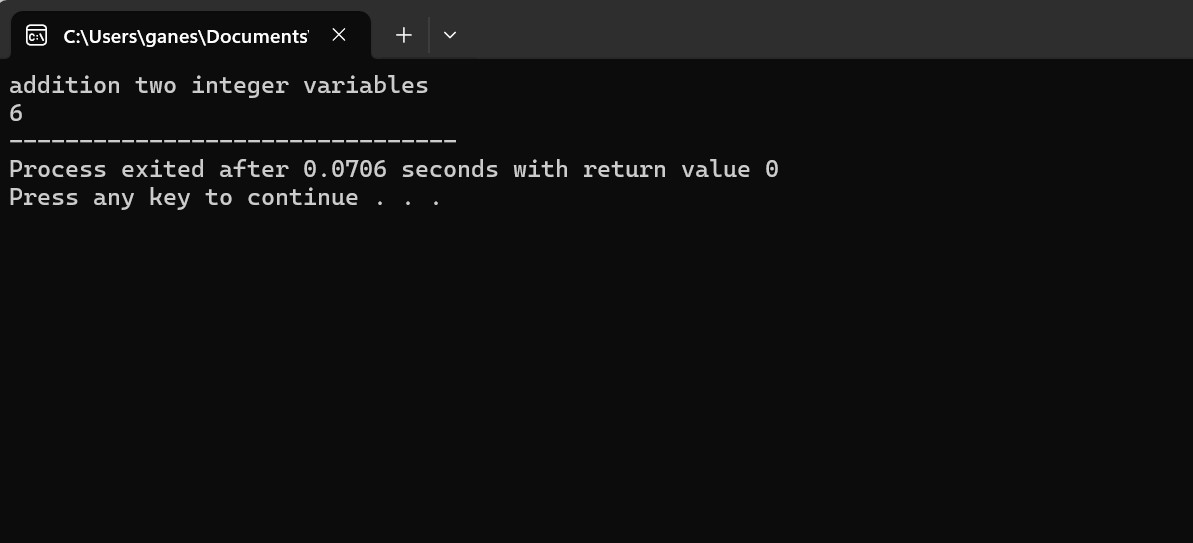
{

int a=2,b=4;

cout << "addition two integer variables"<<endl;

cout <<a+b;

}



6

#include<iostream>

using namespace std;

int main()

{

int a=5;

if (a%2==0)

{

cout <<"even number";

}

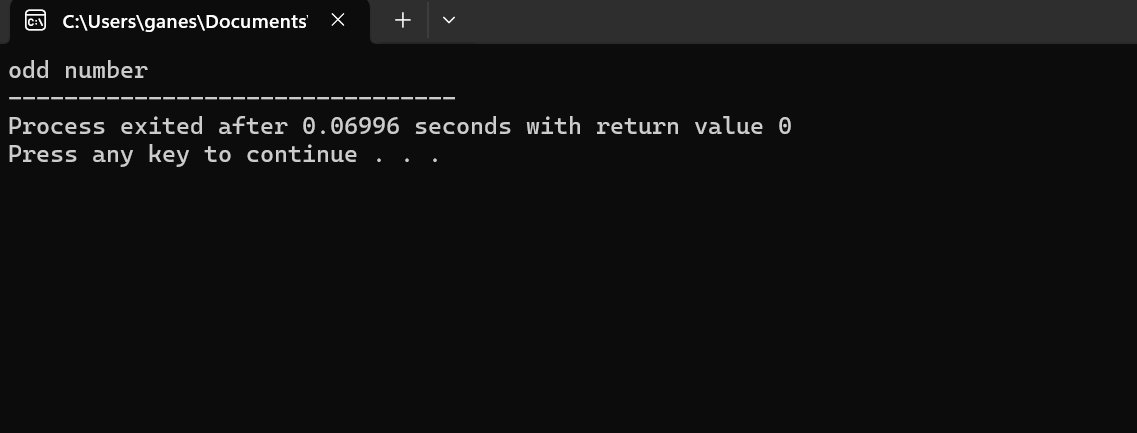
else

{

cout <<"odd number";

}

}



7

#include<iostream>

using namespace std;

int main()

{

int a[5]={1,2,3,4,5};

cout << "the array elements="<<endl;

for(int i=0;i<5;i++)

{

cout << a[i]<<"\n";

}

}

A screenshot of a computer

Description automatically generated

8

#include<iostream>

using namespace std;

int main()

{

int n;

cout <<"enter your choice=";

cin >>n;

switch(n)

{

case 1:

cout<<"sunday";

break;

case 2:

cout <<"monday";

break;

case 3:

cout <<"Tuesday";

break;

case 4:

cout <<"wednesday";

break;

case 5:

cout <<"thursday";

break;

case 6:

cout <<"Friday";

break;

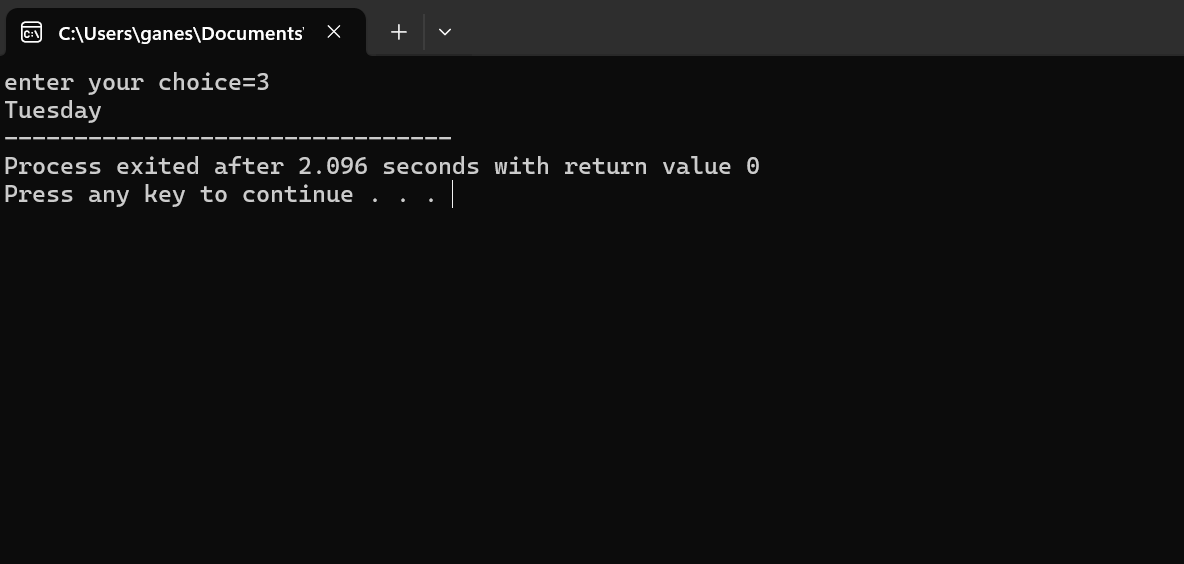
case 7:

cout <<"saturday";

break;

}

}



9

#include<iostream>

using namespace std;

int main()

{

int l,b;

cout <<"enter the length=";

cin >>l;

cout <<"enter the width=";

cin >>b;

int area=l\*b;

cout << "area of rectangle="<<area;

}

A screenshot of a computer

Description automatically generated

10

#include<iostream>

using namespace std;

int main()

{

int a,b,c;

cout <<"enter the three numbers=";

cin >>a >>b>>c;

if ((a>b) && (a>c))

{

cout <<a<<"is greater number"<<endl;

}

else if((b>a)&&(b>c))

{

cout <<b<<"is greater number"<<endl;

}

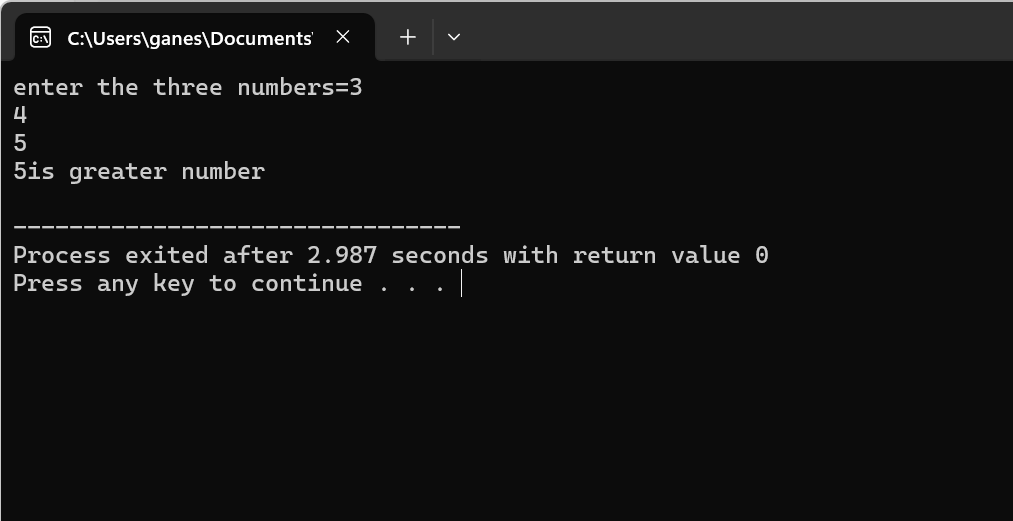
else

{

cout <<c <<"is greater number"<<endl;

}

}



11

#include <iostream>

int main() {

const double PI = 3.14159;

double radius, area;

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

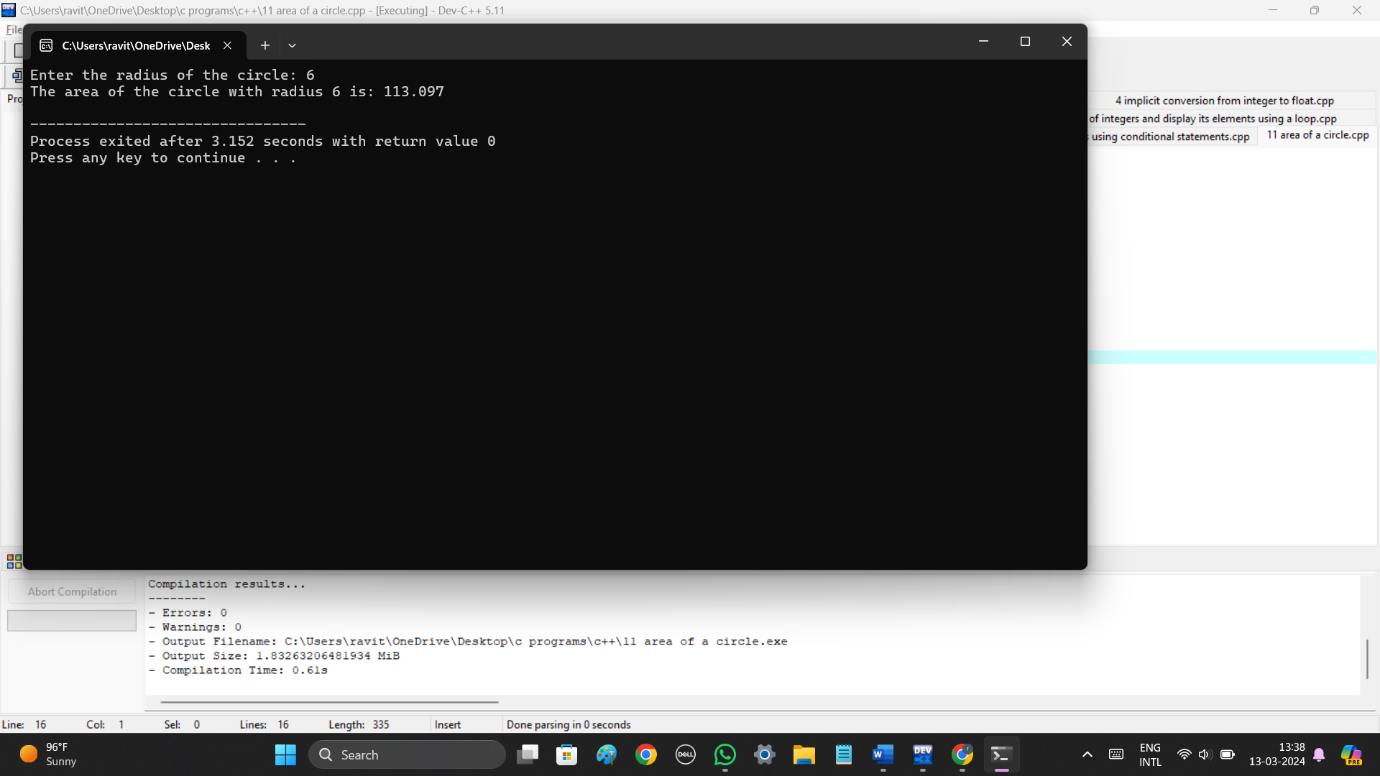
std::cin >> radius;

area = PI \* radius \* radius;

std::cout << "The area of the circle with radius " << radius << " is: " << area << std::endl;

return 0;

}



12

#include <iostream>

int main() {

int intValue;

float floatValue;

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

std::cin >> intValue;

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

std::cin >> floatValue;

float result = floatValue / intValue;

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

return 0;

}

A computer screen with a black screen

Description automatically generated

13

#include <iostream>

int main() {

int year;

bool isLeapYear = false;

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

std::cin >> year;

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

isLeapYear = true;

}

if (isLeapYear) {

std::cout << year << " is a leap year." << std::endl;

} else {

std::cout << year << " is not a leap year." << std::endl;

}

return 0;

}

A computer screen shot of a black screen

Description automatically generated

14

#include <iostream>

int main() {

double length, width, area;

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

std::cin >> length;

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

std::cin >> width;

area = length \* width;

std::cout << "The area of the rectangle is: " << area << std::endl;

return 0;

}

A computer screen with a black screen

Description automatically generated

15

#include <iostream>

int main() {

int number;

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

std::cin >> number;

if (number & 1) {

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

} else {

std::cout << number << " is not an odd number." << std::endl;

}

return 0;

}

A computer screen with a black screen

Description automatically generated

16

#include <iostream>

int main() {

int monthNumber;

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

std::cin >> monthNumber;

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;

default:

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

break;

}

return 0;

A computer screen shot of a black screen

Description automatically generated}

17

#include <iostream>

int main() {

const double PI = 3.14159;

double radius, volume;

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

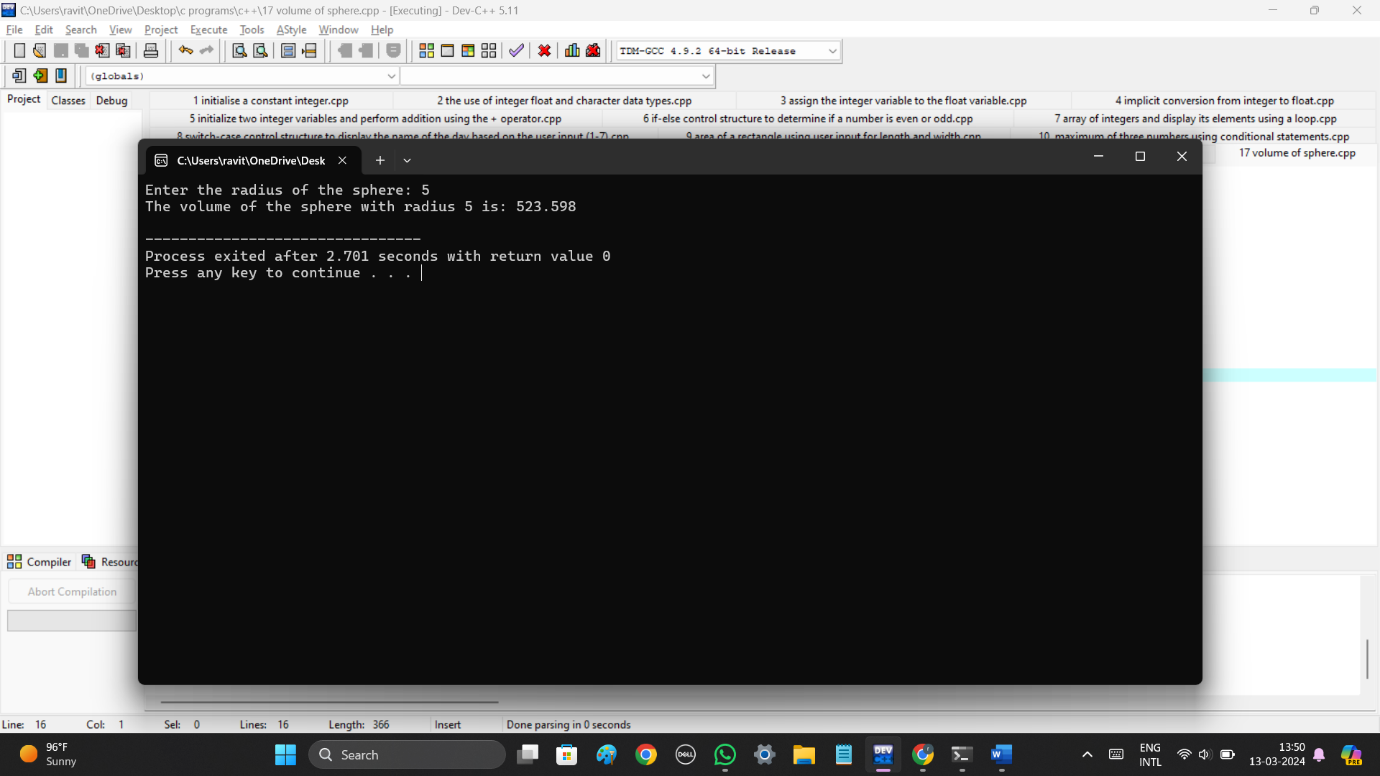
std::cin >> radius;

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

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

return 0;

}



18

#include <iostream>

int main() {

int dividend, divisor;

double result;

std::cout << "Enter the dividend: ";

std::cin >> dividend;

std::cout << "Enter the divisor: ";

std::cin >> divisor;

if (divisor != 0) {

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

std::cout << "Result of division: " << result << std::endl;

} else {

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

}

return 0;

}

A computer screen with a black screen

Description automatically generated

19

#include <iostream>

class Complex {

private:

double real;

double imag;

public:

Complex(double real = 0.0, double imag = 0.0) : real(real), imag(imag) {}

Complex operator+(const Complex& other) {

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

}

Complex operator-(const Complex& other) {

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

}

void display() {

std::cout << real << " + " << imag << "i" << std::endl;

}

};

int main() {

Complex c1(2.0, 3.0);

Complex c2(1.0, 2.0);

Complex sum = c1 + c2;

Complex diff = c1 - c2;

std::cout << "Sum: ";

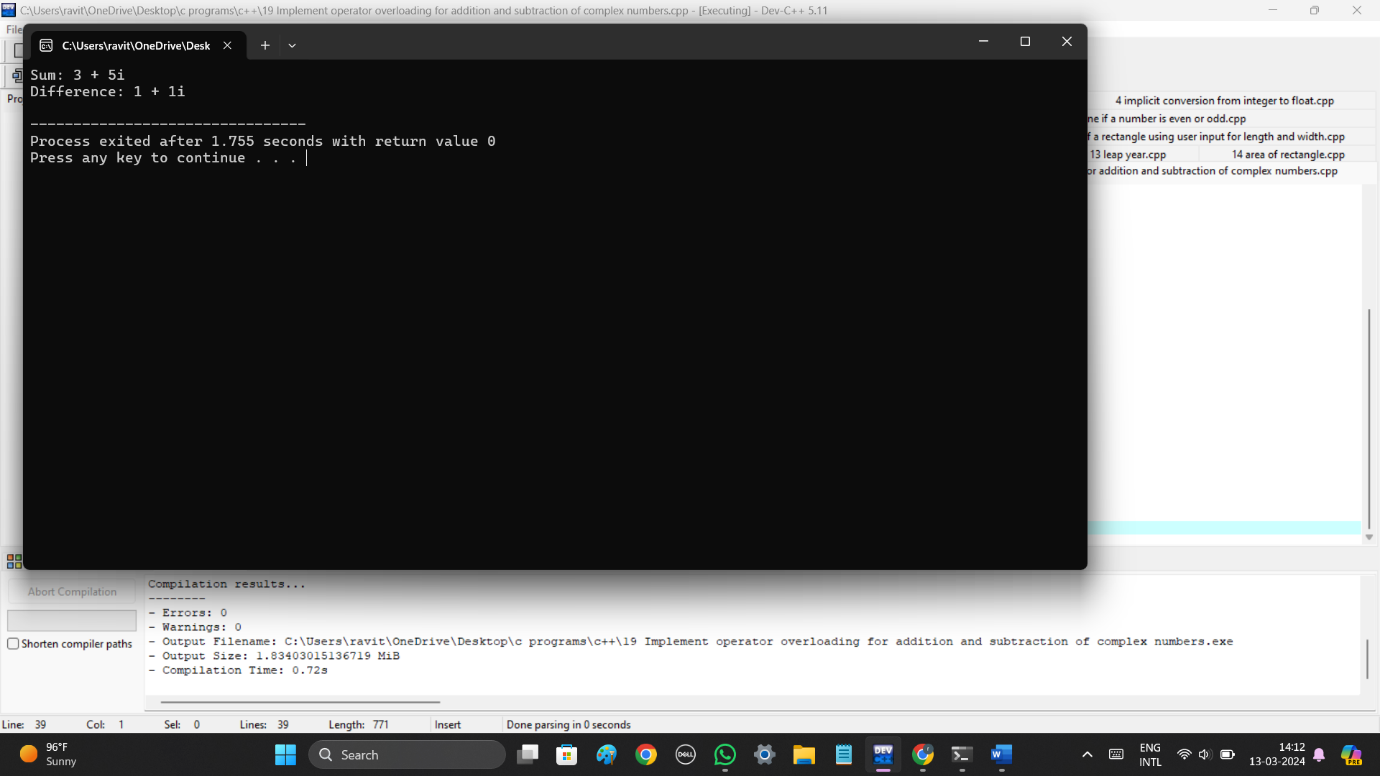
sum.display();

std::cout << "Difference: ";

diff.display();

return 0;

}



20

#include <iostream>

int main() {

int number;

unsigned int factorial = 1;

std::cout << "Enter a positive integer: ";

std::cin >> number;

if (number < 0) {

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

return 1;

}

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

factorial \*= i;

}

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

return 0;

}

A computer screen with a black screen

Description automatically generated

21

#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) {

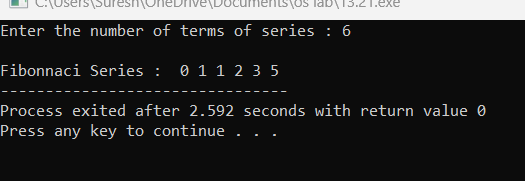
cout << " " << fib(i);

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;

}

A screenshot of a computer

Description automatically generated

24

#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() {

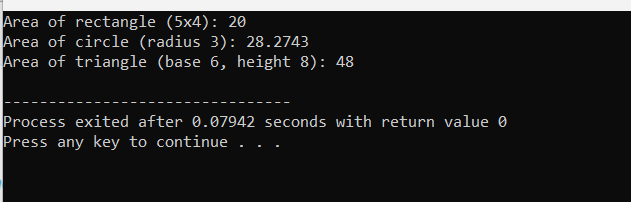
cout << "Area of rectangle (5x4): " << calculateArea(5.0, 4.0) << endl;

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

#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;

}

A screenshot of a computer program

Description automatically generated

27

#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: ";

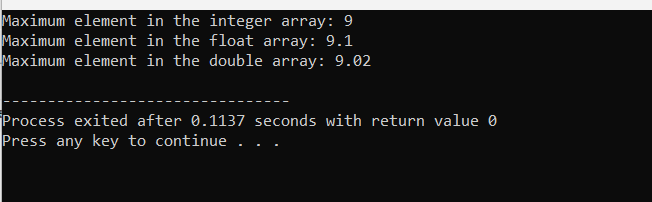
cin >> numDisks;

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

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

return 0;

}



29

#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;

}

A screenshot of a computer

Description automatically generated

30

#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;

}

A screen shot of a computer

Description automatically generated