PROGRAM

DATE:13/03/2024

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

#include <bits/stdc++.h>

using namespace std;

int main(){

int number;

cout<<"enter a constant number";

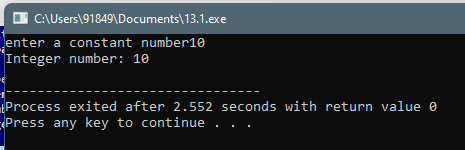
cin>>number;

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

return 0;

}

OUTPUT:



1. 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;

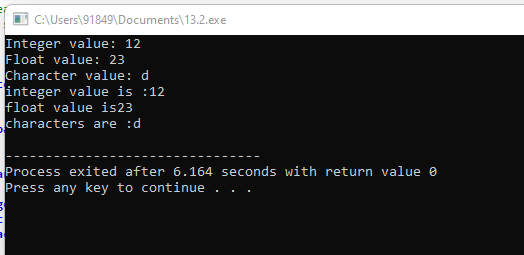
cout<<"float value is"<<m<<endl;

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

return 0;

}

OUTPUT:



1. 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);

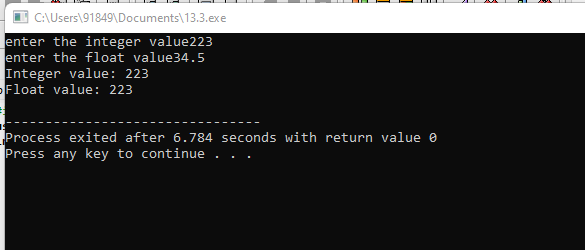
cout << "Integer value: " << n <<endl;

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 value";

cin>>n;

cout<<"enter the float value";

cin>>m;

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

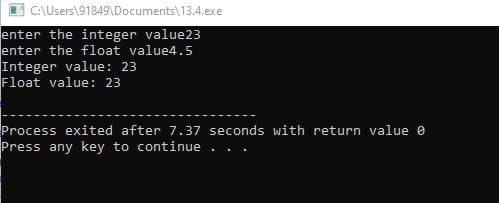
cout << "Integer value: " << n <<endl;

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

return 0;

}

OUTPUT:



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;

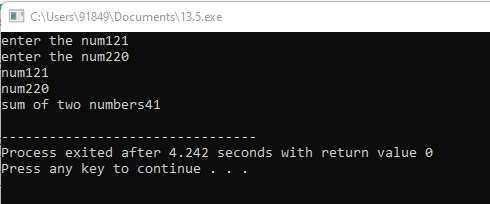
cout<<"num1"<<num1<<endl;

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 n;

cout<<"enter a value";

cin>>n;

if(n%2==0){

cout<<"even";

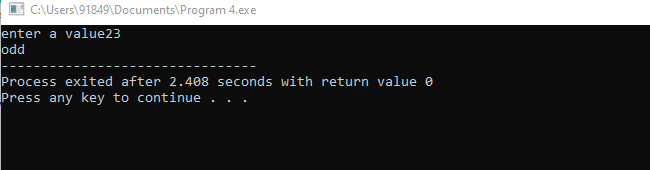
}

else

cout<<"odd";

return 0;

}

OUTPUT:  


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';

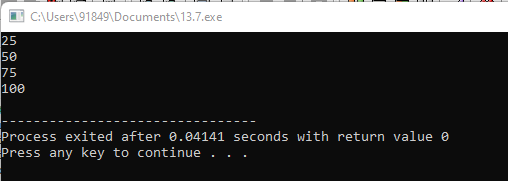
cout << a[1] << '\n';

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

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

return 0;

}

OUTPUT:  


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 << "Enter the day number (1-7): ";

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";

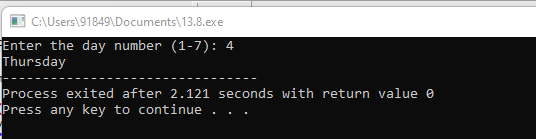
break;

}

return 0;

}

OUTPUT:



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(){

int length, breadth, area;

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

cin >> length;

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

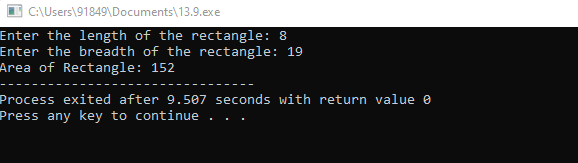
cin >> breadth;

area = length \* breadth;

cout << "Area of Rectangle: " << area;

return 0;

}

OUTPUT:  


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

#include<iostream>

using namespace std;

int main(){

int a,b,c;

cout<<"enter value of a,b,c";

cin>>a>>b>>c;

if(a>b&&a>c)

{

cout<<"a is maximum";

}

else if(b>c)

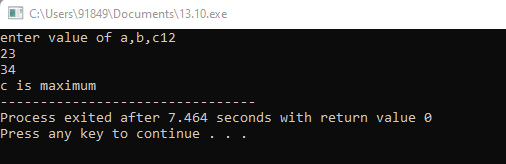
cout<<"b is maximum";

else

cout<<"c is maximum";

return 0;

}



MEDIUM

1. 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: ";

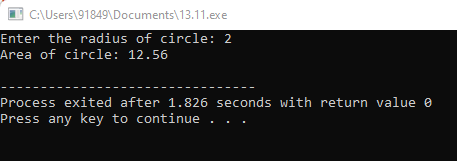
cin >> radius;

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 {

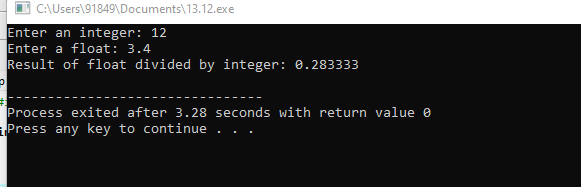
std::cout << "Error: Division by zero is not allowed." << std::endl;

}

return 0;

}

OUTPUT:



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) {

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

}

else if (year % 100 == 0) {

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

}

else if (year % 4 == 0) {

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

}

else {

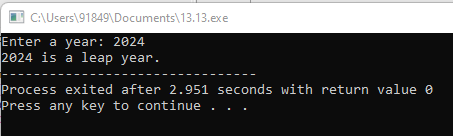
cout << year << " is not a leap year.";

}

return 0;

}

OUTPUT:



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, breadth, area;

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

cin >> length;

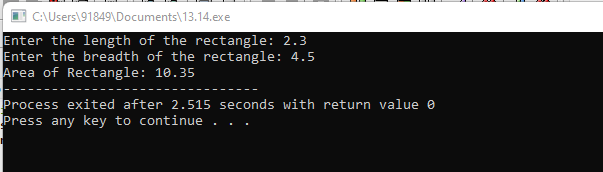
cout << "Enter the breadth of the rectangle: ";

cin >> breadth;

area = length \* breadth;

cout << "Area of Rectangle: " << area;

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>

using namespace std;

int main() {

int n;

cin>>n;

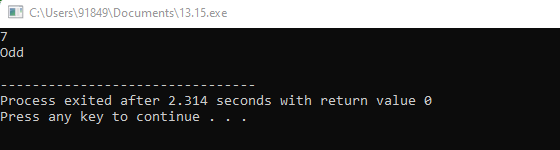
if (n&1) cout <<"Odd"<<endl;

else cout <<"Even"<<endl;

return 0;

}

OUTPUT



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>

using namespace std;

int main()

{

int month;

cout << "Enter the number of month in which you are born : " << endl;

cin >> month;

switch (month)

{

case 1:

cout << "January is the month";

break;

case 2:

cout << "February is the month";

break;

case 3:

cout << "March is the month";

break;

case 4:

cout << "April is the month";

break;

case 5:

cout << "May is the month";

break;

case 6:

cout << "June is the month";

break;

case 7:

cout << "July is the month";

break;

case 8:

cout << "August is the month";

break;

case 9:

cout << "September is the month";

break;

case 10:

cout << "October is the month";

break;

case 11:

cout << "November is the month";

break;

case 12:

cout << "December is the month";

break;

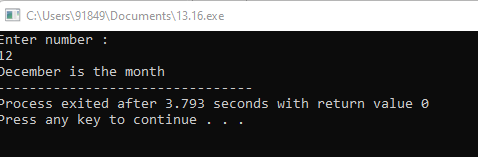
default:

cout << "Error, Enter the correct value ";

}

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()

{

int rad1;

float vol;

cout << " Input the radius of a sphere : ";

cin >> rad1;

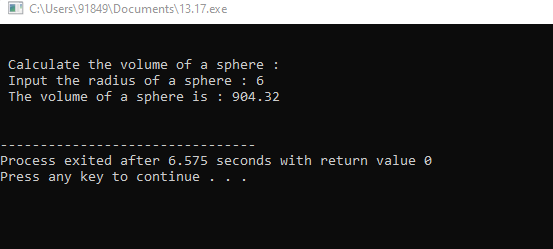
vol = (4 \* 3.14 \* rad1 \* rad1 \* rad1) / 3;

cout << " The volume of a sphere is : "<< volsp << endl;

cout << 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 a, b;

cout << "Enter the numerator: ";

cin >> a;

cout << "Enter the denominator: ";

cin >> b;

if (b == 0) {

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

} else {

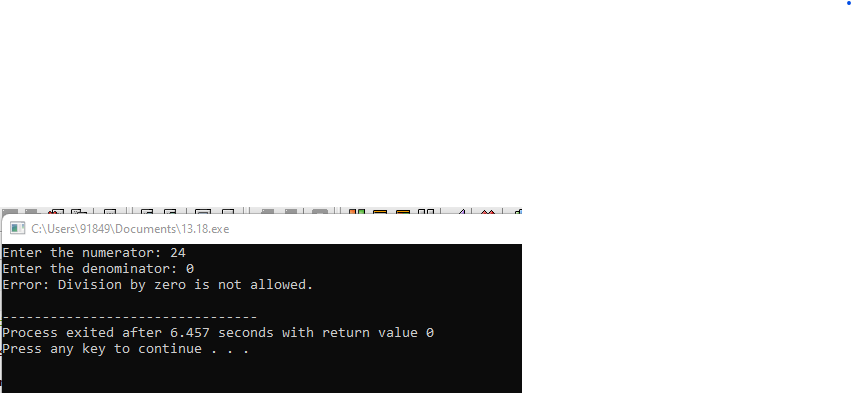
double result = static\_cast<double>(a) / b;

cout << "Result of division: " << result << 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>

#include <sstream>

#include <cmath>

using namespace std;

class Complex {

private:

int real, imag;

public:

Complex(){

real = imag = 0;

}

Complex (int r, int i){

real = r;

imag = i;

}

string to\_string(){

stringstream ss;

if(imag >= 0)

ss << "(" << real << " + " << imag << "i)";

else

ss << "(" << real << " - " << abs(imag) << "i)";

return ss.str();

}

Complex operator+(Complex c2){

Complex ret;

ret.real = real + c2.real;

ret.imag = imag + c2.imag;

return ret;

}

};

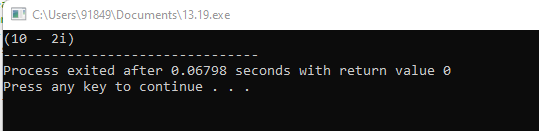
int main(){

Complex c1(8,-5), c2(2,3);

Complex res = c1 + c2;

cout << res.to\_string();

}



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>

using namespace std;

int main()

{

int n;

int fact=1;

cout << "Enter number : " << endl;

cin >> n;

if(n>0) {

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

fact \*= i;

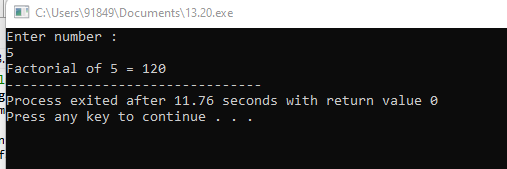
}

cout << "Factorial of " << n << " = " << fact;

}

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 main() {

int n1=0,n2=1,n3,i,n;

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

cin>>n;

cout<<n1<<" "<<n2<<" ";

for(i=2;i<n;++i)

{

n3=n1+n2;

cout<<n3<<" ";

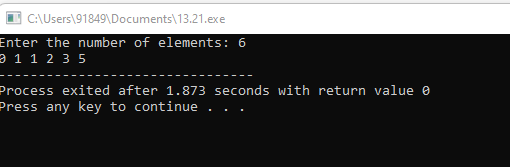
n1=n2;

n2=n3;

}

return 0;

}



22. Develop a C++ program that dynamically allocates memory for an array of integers based on user input for the array size. Implement error handling for memory allocation failure.

23. Create a C++ program that defines a class representing a matrix. Implement operator overloading for matrix addition, subtraction, and multiplication.

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>

using namespace std;

int area(int);

int area(int,int);

float area(float);

float area(float,float);

int main()

{

int s,l,b;

float r,bs,ht;

cout<<"Enter side of a square:";

cin>>s;

cout<<"Enter length and breadth of rectangle:";

cin>>l>>b;

cout<<"Enter radius of circle:";

cin>>r;

cout<<"Enter base and height of triangle:";

cin>>bs>>ht;

cout<<"Area of square is"<<area(s);

cout<<"\nArea of rectangle is "<<area(l,b);

cout<<"\nArea of circle is "<<area(r);

cout<<"\nArea of triangle is "<<area(bs,ht);

}

int area(int s)

{

return(s\*s);

}

int area(int l,int b)

{

return(l\*b);

}

float area(float r)

{

return(3.14\*r\*r);

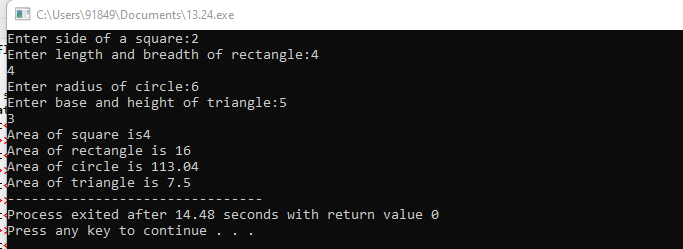
}

float area(float bs,float ht)

{

return((bs\*ht)/2);

}



25. Develop a C++ program that evaluates mathematical expressions entered by the user, considering operator precedence and associativity. Support parentheses for grouping.

#include <iostream>

using namespace std;

int main() {

int num1 = 5 - 17 \* 6;

int num2 = 5 - (17 \* 6);

int num3 = (5 - 17) \* 6;

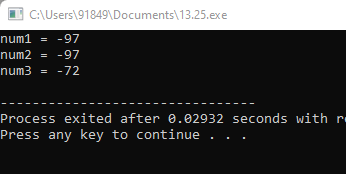
cout << "num1 = " << num1 << endl;

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

cout << "num3 = " << num3 << 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 add(T num1, T num2) {

return (num1 + num2);

}

int main() {

int result1;

double result2;

result1 = add<int>(2, 3);

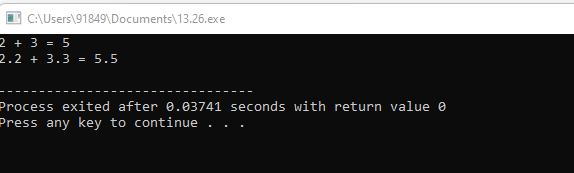
cout << "2 + 3 = " << result1 << endl;

result2 = add<double>(2.2, 3.3);

cout << "2.2 + 3.3 = " << result2 << 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 TOH(int n, char Sour, char Aux, char Des)

{

if (n == 1) {

cout << "Move Disk " << n << " from " << Sour << " to " << Des << endl;

return;

}

TOH(n - 1, Sour, Des, Aux);

cout << "Move Disk " << n << " from " << Sour << " to " << Des << endl;

TOH(n - 1, Aux, Sour, Des);

}

int main()

{

int n;

cout << "Enter no. of disks:";

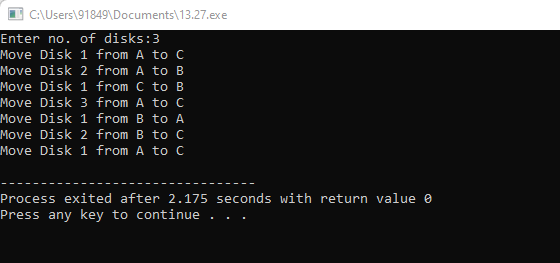
cin >> n;

TOH(n, 'A', 'B', 'C');

return 0;

}

OUTPUT:



28. . Develop a C++ program that reads input from a file containing numeric data and calculates the average. Implement exception handling to handle file I/O errors and invalid data.

#include <iostream>

#include <stdexcept>

using namespace std;

int main()

{

try {

int numerator = 10;

int denominator = 0;

int res;

if (denominator == 0) {

throw runtime\_error(

"Division by zero not allowed!");

}

res = numerator / denominator;

cout << "Result after division: " << res << endl;

}

catch (const exception& e) {

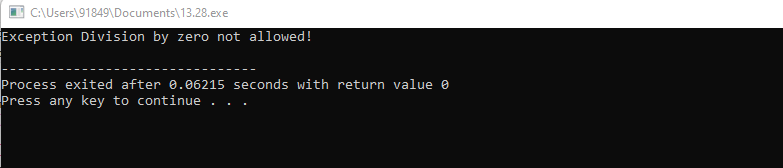
cout << "Exception " << e.what() << endl;

}

return 0;

}

OUTPUT:



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;

int foo(char c) { return (int)c; }

void printASCIIcode(char c, int(\*func\_ptr)(char))

{

int ascii = func\_ptr(c);

cout << "ASCII code of " << c << " is: " << ascii;

}

int main()

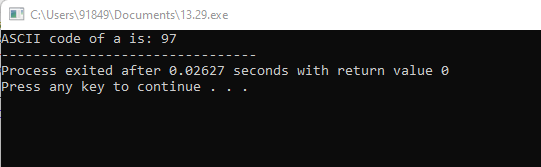
{

printASCIIcode('a', &foo);

return 0;

}

OUTPUT:



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;

}

OUTPUT:

