**LAB 2:WRITING A C++ PROGRAM USING CLASSES AND OBJECTS.**

**OBJECTIVE:**

1. Understand the fundamental concepts of classes and objects in object-oriented programming.
2. Learn how to create and define classes.
3. Explore the concept of encapsulation and how it is implemented using classes.

**THEORY:**

Object-oriented programming(OOP)is a powerful and widely adopted programming paradigm that fundamentally transforms the way software is designed and developed. At its core lies the concept of classes and objects,which form the comerstone of OOP’s modularity,reusability,and abstraction.

Classes are the elemental building blocks of OOP that defines the structure and behaviour of objects.A class serves a blueprint or template that encapsulates a set of attributes(properties or fields) and methods(functions)that correctly represent an abstract entity.A class is a template definition of the methods and variables in a particular kind of object.And an object is a specific instance of a class.

A class defines constituent members which enable these class instances to have state and behaviour.Also we can say that class is an user defined data types

that contains members variables and member functions.

**1.Define a class Person with private members for the person’s name and age.Write methods to set and get these values.**

**SOURCE CODE:**

#include<iostream>

using namespace std;

class Person{

private:

string name;

int age;

public:

void setData(string n , int a){

name=n;

age=a;

}

void getData(){

cout<<"My name is " <<name<<endl;

cout<<"I am " <<age << " years old ."<<endl;

}

};

int main(){

Person p;

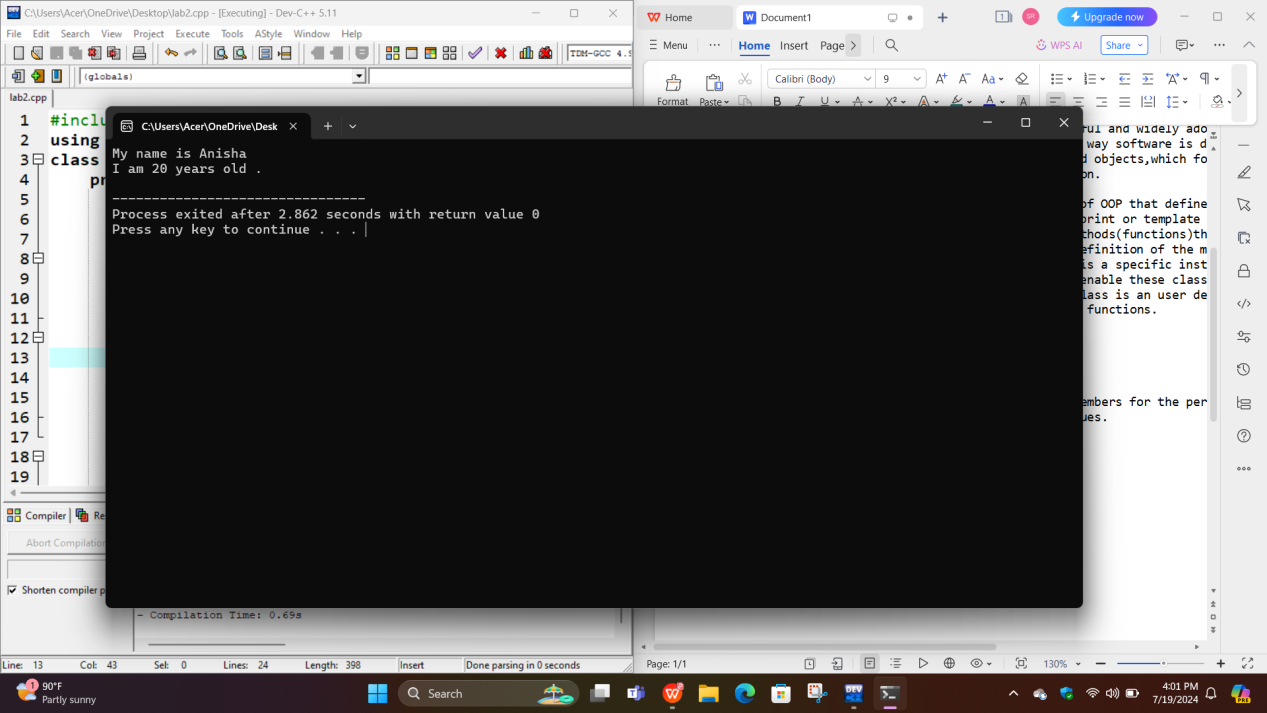
p.setData("Anisha" , 20);

p.getData();

return 0;

}

**OUTPUT:**

****

**2.Create a class Point that represents a point in 2D space with x and y coordinates.Write methods to set and get the coordinates.**

**//SOURCE CODE:**

#include <iostream>

using namespace std;

class Point {

private:

int x, y;

public:

void setCoordinates(int xCord, int yCord) {

x = xCord;

y = yCord;

}

void getCoordinates(){

cout<<"X coordinate is "<<x<<endl;

cout<<"Y coordinate is "<<y<<endl;

}

};

int main() {

Point p1;

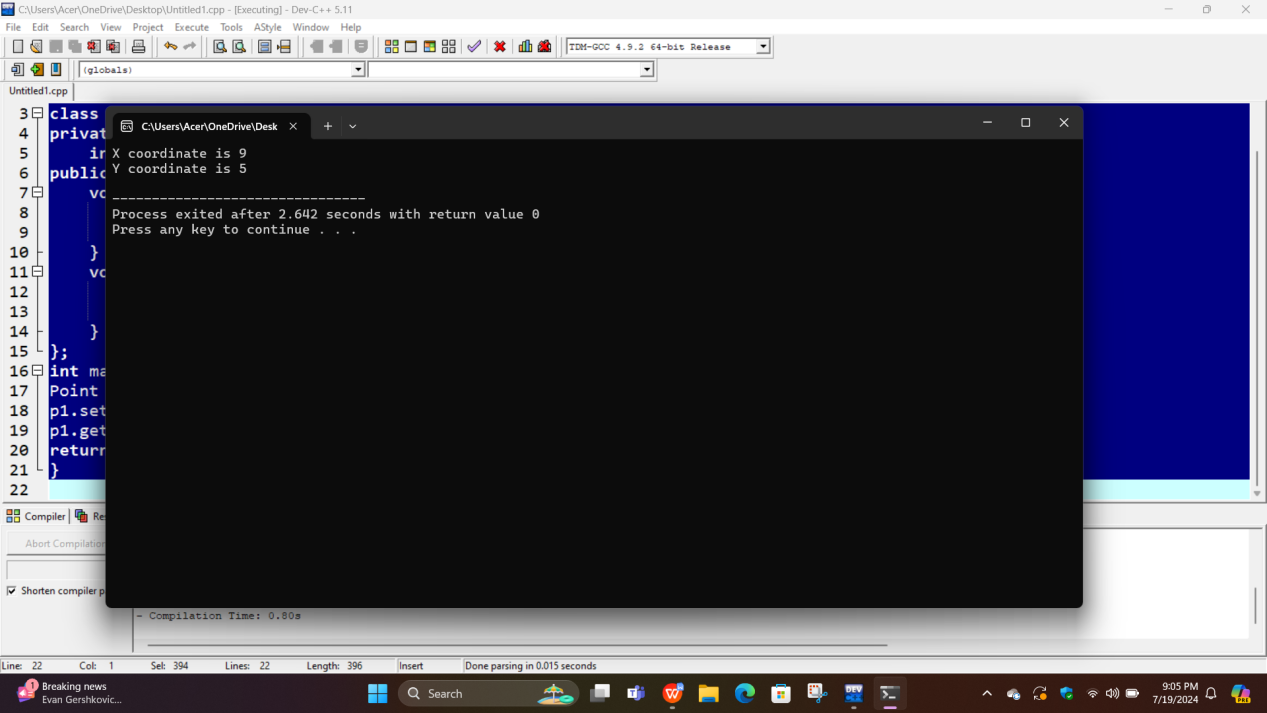
p1.setCoordinates(9,5);

p1.getCoordinates();

return 0;

}

**OUTPUT:**

****

**3.Define a class Circle with a member for the radius.Write a methods to calculate the circumference and area of circle.**

**//SOURCE CODE:**

#include<iostream>

#define PI 3.14

using namespace std;

class Circle{

private:

int radius;

public:

void setData( int r){

radius=r;}

void getData(){

cout<<"The area of circle is "<< PI\*radius\*radius<<endl;

cout<<"The circumference of circle is "<< 2\* PI \*radius\*radius;}

};

int main(){

Circle c;

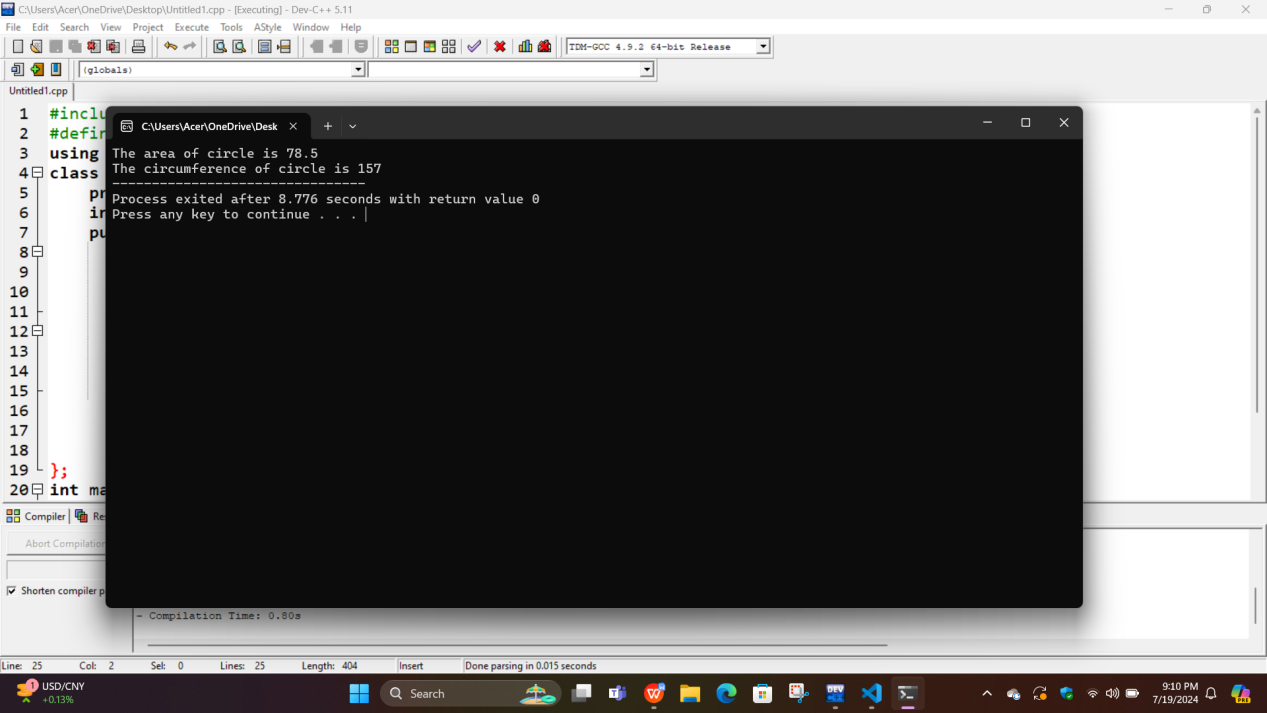
c.setData(5);

c.getData();

return 0;

}

**OUTPUT:**

****

**4.Implement a class Book with members for the title,author,price and number pf pages.Include the methods to set and get these values.**

**SOURCE CODE:**

#include<iostream>

using namespace std;

class Book{

private:

string title;

string author;

float price;

int pagenumber;

public:

void setData( string t , string m , float p, int n){

title=t;

author=m;

price=p;

pagenumber=n;

}

void getData(){

cout<<"Title of the book is "<<title<<endl;

cout<<"Author of the book is "<<author<<endl;

cout<<"Price of the book is Rs "<<price<<endl;

cout<<"Number of pages in this book is "<<pagenumber<<endl;

}

};

int main(){

Book b;

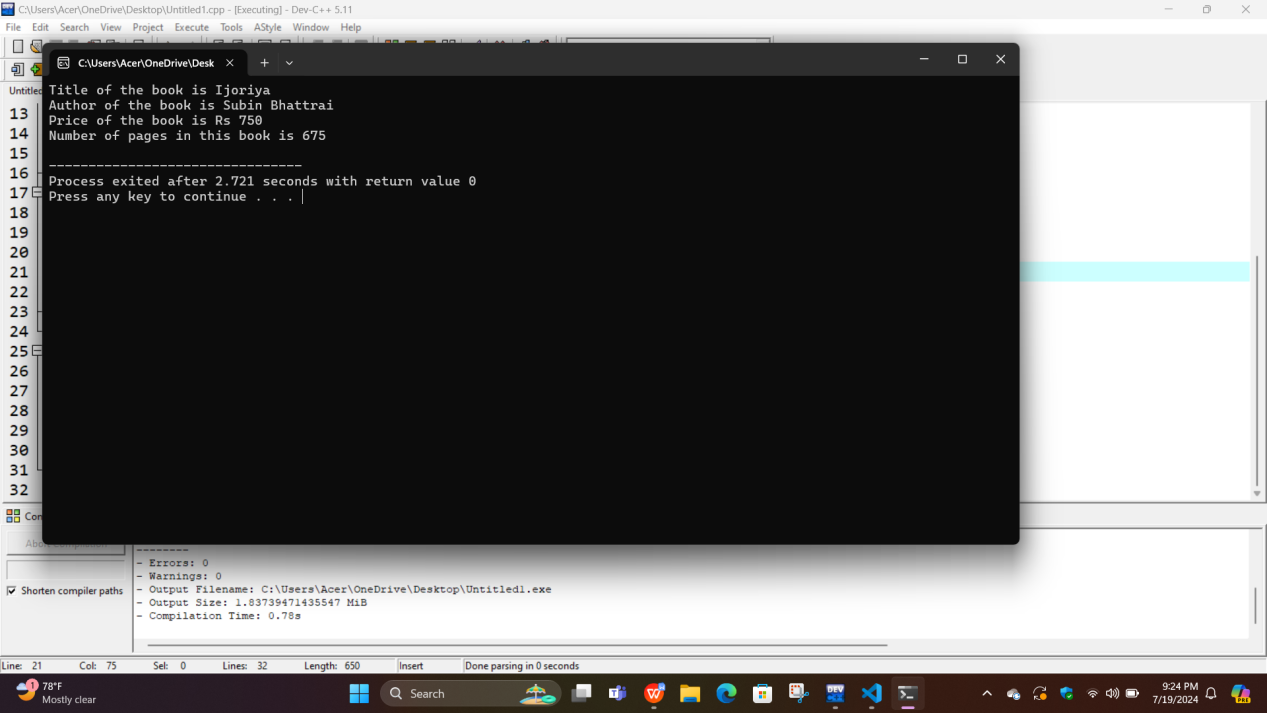
b.setData("Ijoriya" , "Subin Bhattrai", 750, 675 );

b.getData();

return 0;

}

**OUTPUT:**

****

**5.Create a class BankAccount with members for the account number and balance. Write the methods to deposit and withdraw money,and to check the balance.**

**SOURCE CODE:**

#include <iostream>

#include <string>

using namespace std;

class BankAccount {

private:

string accountNumber;

double balance;

public:

BankAccount(string accNumber, double initialBalance) {

accountNumber = accNumber;

balance = initialBalance;

}

void deposit(double amount) {

if (amount > 0) {

balance += amount;

cout << "Successfully deposited Rs" << amount << ". Finally i have got Rs " << balance << endl;

} else {

cout << "Deposit amount must be positive." << endl;

}

}

void withdraw(double amount) {

if (amount > 0) {

if (balance >= amount) {

balance -= amount;

cout << "Successfully withdrew Rs" << amount << ". New balance is Rs" << balance << endl;

} else {

cout << "Sorry!! Looking at your balance you seems poor. You cannot withdraw Rs " << amount << endl;

}

}

}

string getAccountNumber() {

return accountNumber;

}

double getBalance() {

return balance;

}

};

int main() {

BankAccount account( "0975342568" , 1000.0);

cout << "My account number is :" << account.getAccountNumber() << endl;

cout << "Current balance: Rs" << account.getBalance() << endl;

account.deposit(500.0);

account.withdraw(15000.0);

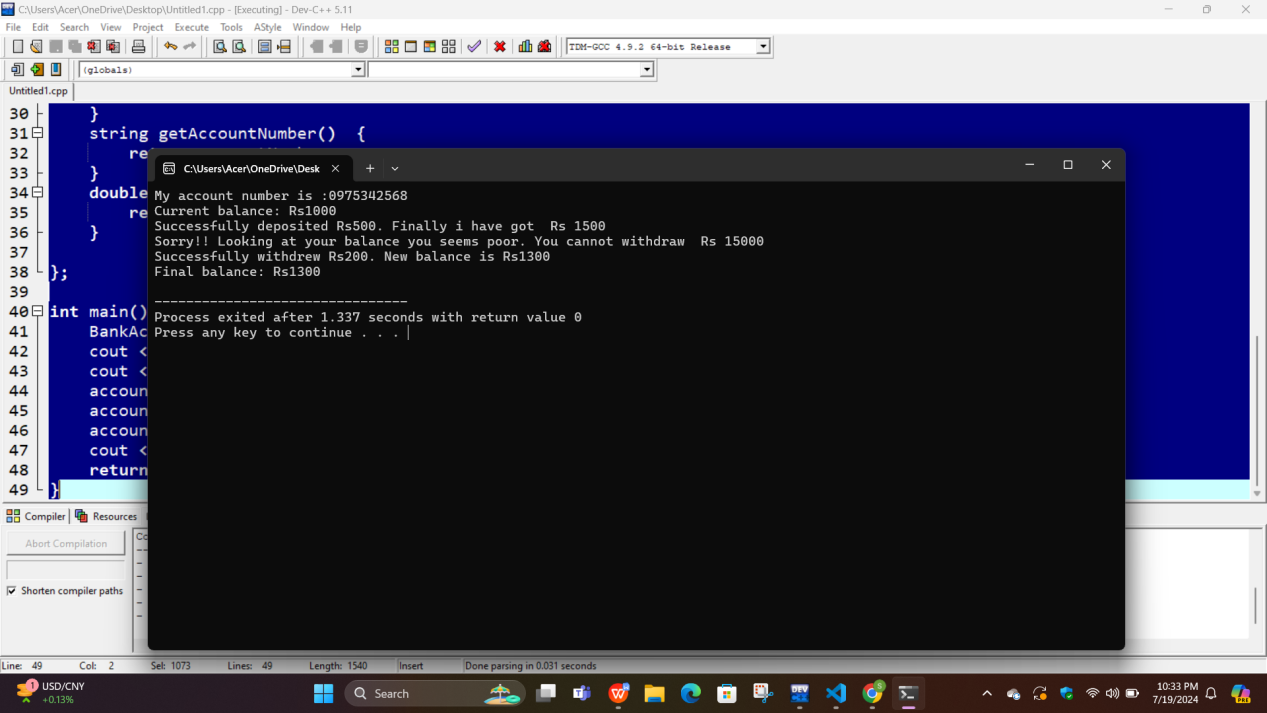
account.withdraw(200.0);

cout << "Final balance: Rs" << account.getBalance() << endl;

return 0;

}

**OUTPUT:**

****

1. **Write a class Date that represents a date with day, month,and year members.Include methods to set and display the date.**

//SOURCE CODE

#include <iostream>

using namespace std;

class Date{

private:

int day;

int month;

int year;

public:

void setDate(int d, int m, int y) {

day = d;

month = m;

year = y;

}

void displayDate(){

cout<<"Today date is:"<<day<<"/" << month<<"/" << year <<endl;

}

};

int main(){

Date d1;

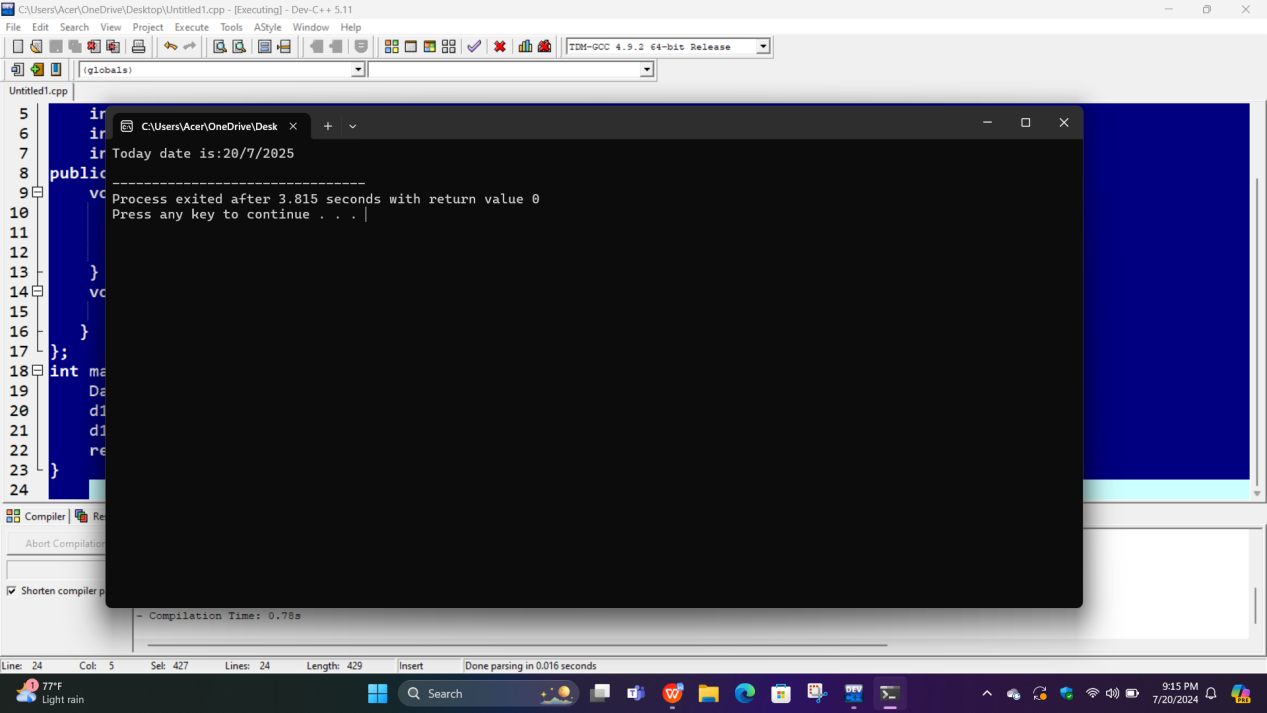
d1.setDate(20, 7, 2025);

d1.displayDate();

return 0;

}

**OUTPUT:**

****

**7.Create a class ComplexNumber and write methods to set and get the number.**

**//SOURCE CODE**

#include<iostream>

using namespace std;

class ComplexNumber{

int a, b;

public:

void setNumber(int n1, int n2){

a = n1;

b = n2;

}

void getNumber(){

cout<<"Your complex number is "<<a<<" + "<<b<<"i"<<endl;

}

};

int main(){

ComplexNumber c1;

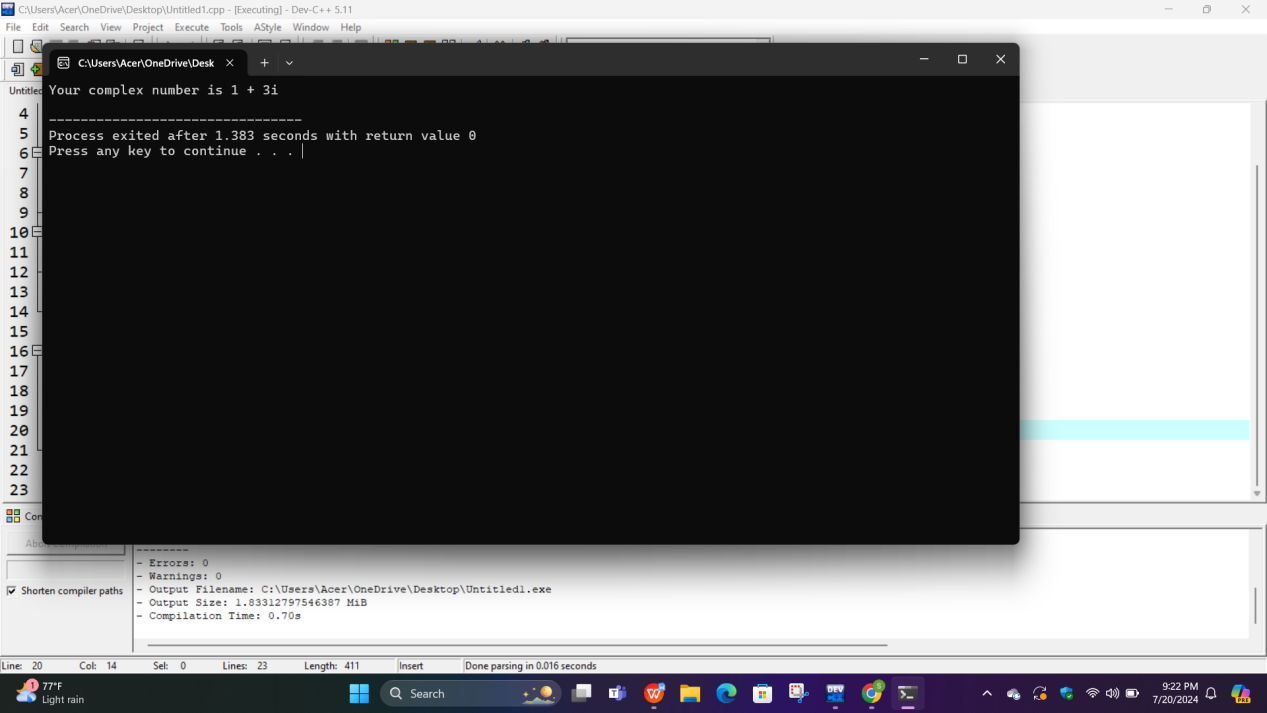
c1.setNumber(1, 3);

c1.getNumber();

return 0;

}

**OUTPUT:**

****

**CONCLUSION:**

In this lab we have covered the essential concepts of classes and objects.We learned that classes acts as templates for creating objects,encapsulating both data and methods.Object creation allows us to represent real-world entities as individual instances.