Lab No: 2

Writing C++ program using Classes and Objects

In C++, a class serves as a blueprint for generating objects, containing data members (variables)and member functions (methods). An object is an instance of a class, holding its own copies of the class data members. Classes outline the structure for objects by defining their properties and behaviors, whereas objects are concrete instances with actual data. This framework, inheritance and polymorphism, which are fundamental principles of object-oriented programming.

Program: 1

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

#include<iostream>

using namespace std;

class person{

private:

string name;

int age;

public:

void get(){

cout<<"Enter your name and age"<<endl;

cin>>name>>age;

}

void set(){

cout<<"name"<<name<<endl;

cout<<"age"<<age<<endl;

}

};

int main(){

person w1,w2;

w1.get();

w2.get();

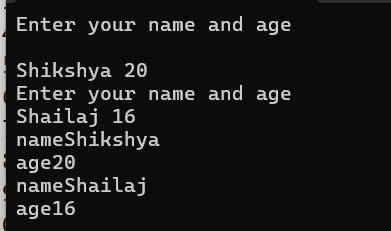
w1.set();

w2.set();

return 0;

}

OUTPUT



Program 2:

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

#include<iostream>

using namespace std;

class Point {

private:

int x;

int y;

public:

Point(int xValue, int yValue) : x(xValue), y(yValue) {}

void getCords() {

cout<<"X-coordinate:"<<x<<endl;

cout<<"Y-coordinate:"<<y<<endl;

}

};

int main() {

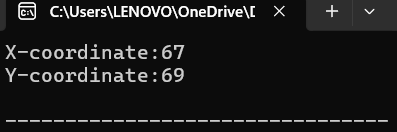
Point p(67,69);

p.getCords();

return 0;

}

OUTPUT:



Program :3

Define a class Circle with a member radius. Write a method to calculate circumference and area of circle.

#include <iostream>

using namespace std;

class Circle{

private:

int radius;

const double PI = 3.14;

public:

void set\_radius(int r){

radius = r;

}

double get\_circum(){

double circum = 2\*PI\*radius;

return circum;

}

double get\_area(){

double area = PI\*radius\*radius;

return area;

}

};

int main()

{

Circle cx, cy;

cx.set\_radius(10);

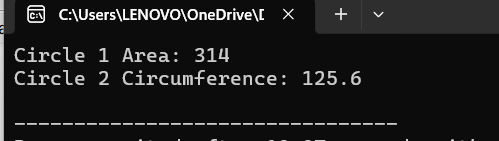
cy.set\_radius(20);

cout << "Circle 1 Area: "<< cx.get\_area()<<endl;

cout << "Circle 2 Circumference: "<< cy.get\_circum()<<endl;

return 0;

}



Program: 4

Implement a class Book with member for title,author,price and pages. Write methods to set and get the values.

#include<iostream>

using namespace std;

class Book {

private:

string title;

string author;

float price;

int pages;

public:

Book(const string&t, const string&a, float p, int pg)

: title(t), author(a), price(p), pages(pg) {};

string getTitle() {

return title;

}

string getAuthor() {

return author;

}

float getPrice() {

return price;

}

int getPages() {

return pages;

}

};

int main(){

Book b1("It ends with us","Coolen Hoover",890.33,400);

cout<<"Title:"<<b1.getTitle()<<endl;

cout<<"Author:"<<b1.getAuthor()<<endl;

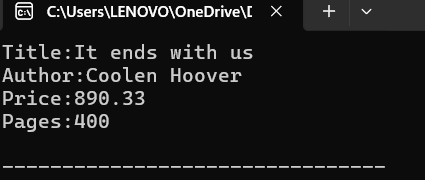
cout<<"Price:"<<b1.getPrice()<<endl;

cout<<"Pages:"<<b1.getPages()<<endl;

return 0;

}

OUTPUT:



Program:5

write a class Date that represents a date with day,month,year .Include method to get and set values.

#include<iostream>

using namespace std;

class Date {

private:

int day;

int month;

int year;

public:

Date(int d, int m, int y) : day(d), month(m), year(y) {}

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

day = d;

month = m;

year =y;

cout<<"Updateddate:"<<day<<""<<"month:"<<month<<""<<"year:"<<year<<endl;

}

void getDate(){

cout<<day<<"/"<<month<<"/"<<year<<endl;

}

};

int main() {

Date today(24, 17, 2005);

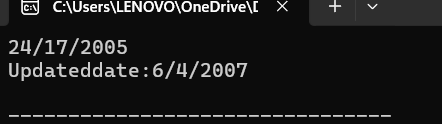
today.getDate();

today.updateDate(6,4,2007);

return 0;

}

OUTPUT:



Program: 6

Create class Complex Number and write the method to get and set the values.

#include<iostream>

using namespace std;

class Complex{

private:

float real;

float imaginary;

public:

Complex(float r = 0.0, float i = 0.0) : real(r), imaginary(i) {}

void setReal(float r) {

real = r;

}

void setImaginary(float i) {

imaginary = i;

}

void get(){

cout<<real<<"+"<<imaginary<<"i"<<endl;

}

};

int main() {

Complex c1(8.0, 9.0);

c1.get();

c1.setReal(4);

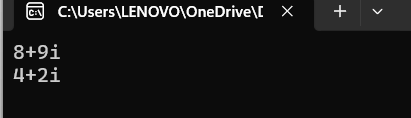
c1.setImaginary(2);

c1.get();

 return 0;

}

OUTPUT:



CONCLUSION:

Therefore class and objects helps in code reusability which makes code easier to understand,maintain and debug.

LAB:3

OBJECTIVE : To illustrate the concept of different types of constructors.

THEORY:

In C++, constructors are special member functions that are automatically called when an object of a class is created. They are used to initialize objects of their class. Here are the types of constructors in C++:

1. **Default Constructor**:
   * A constructor that doesn’t take any arguments is called a default constructor.
   * It initializes the object with default values (if any are specified).

SYNTAX:

class MyClass

{

public: MyClass()

{// Default constructor

}

};

2. **Parameterized Constructor**:

* A constructor that takes parameters is called a parameterized constructor.
* It allows you to initialize an object with user-defined values.

SYNTAX:

class MyClass {

public:

int x;

MyClass(int val) {

x = val;

}

};

**3.Copy Constructor**:

* A constructor which creates a new object as a copy of an existing object is called a copy constructor.
* It is used to initialize an object using another object of the same class.

SYNTAX:

class MyClass {

public:

int x;

MyClass(const MyClass &obj) {

x = obj.x;

}

};

PROGRAM 1:

Write a program to demonstrate the use of different types of constructors in c++

#include<iostream>

using namespace std;

class complex{

float real;

float imag;

public:

complex(){ //default constructor

real=0;

imag=0;

}

complex(float r,float i) //parameterized constructor

{

real=r;

imag=i;

}

complex(complex &c) //copy constructor

{

real=c.real;

imag=c.imag;

}

void showdata(){

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

}

};

int main(){

complex c1;

complex c2(4.3,5.6);

complex c3(c2);

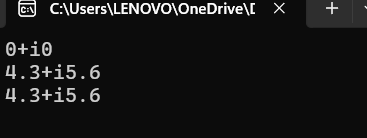
c1.showdata();

c2.showdata();

c3.showdata();

return 0;

}



PROGRAM 2:

Write a class person with a constructor that initializes the name and age of the person.

#include<iostream>

using namepsace std;

class person{

private:

string name;

int age;

public:

person(string n,int a):name(n),age(a){};

void info(){

cout<<"Name:"<<name<<endl;

cout<<"Age:"<<age<<endl;

}

};

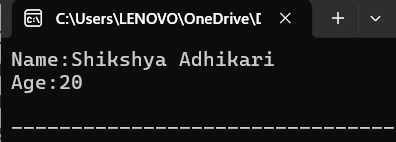
int main(){

person m1("Shikshya Adhikari",20);

p1.info();

return 0;

}



Program 7:

Calculate the class bank account with members account number and check the deposit,withdraw and current balance.

#include<iostream>

using namespace std;

class BankAccount {

private:

int accountNumber;

float balance;

public:

BankAccount(int accNum, float initialBalance):accountNumber(accNum), balance(initialBalance) {}

void deposit(float amount) {

cout<<amount<<"was deposited"<<endl;

checkBal();

balance += amount;

}

void withdraw(float amount) {

cout<<amount<<"was withdrawn"<<endl;

balance -= amount;

}

void checkBal(){

cout<<"your current balance is"<<balance<<endl;

}

};

int main() {

BankAccount myAccount(987654321, 7652);

myAccount.deposit(900.00);

myAccount.withdraw(200.00);

return 0;

}

