**Assignment 8**

A1. Define a class and declare some member variables and member functions.

**Code :**

**#include <iostream>**

**using namespace std;**

**class Box {**

**public:**

**double length;**

**double breadth;**

**double height;**

**double getVolume(void);**

**void setLength( double len );**

**void setBreadth( double br );**

**void setHeight( double he );**

**};**

**double Box::getVolume(void) {**

**return length \* breadth \* height;**

**}**

**void Box::setLength( double len ) {**

**length = len;**

**}**

**void Box::setBreadth( double br ) {**

**breadth = br;**

**}**

**void Box::setHeight( double he ) {**

**height = he;**

**}**

**int main() {**

**Box Box1;**

**Box Box2;**

**double volume = 0.0;**

**Box1.setLength(7.0);**

**Box1.setBreadth(5);**

**Box1.setHeight(6.0);**

**Box2.setLength(24.0);**

**Box2.setBreadth(15.0);**

**Box2.setHeight(8.0);**

**volume = Box1.getVolume();**

**cout << "Volume of Box1 : " << volume <<endl;**

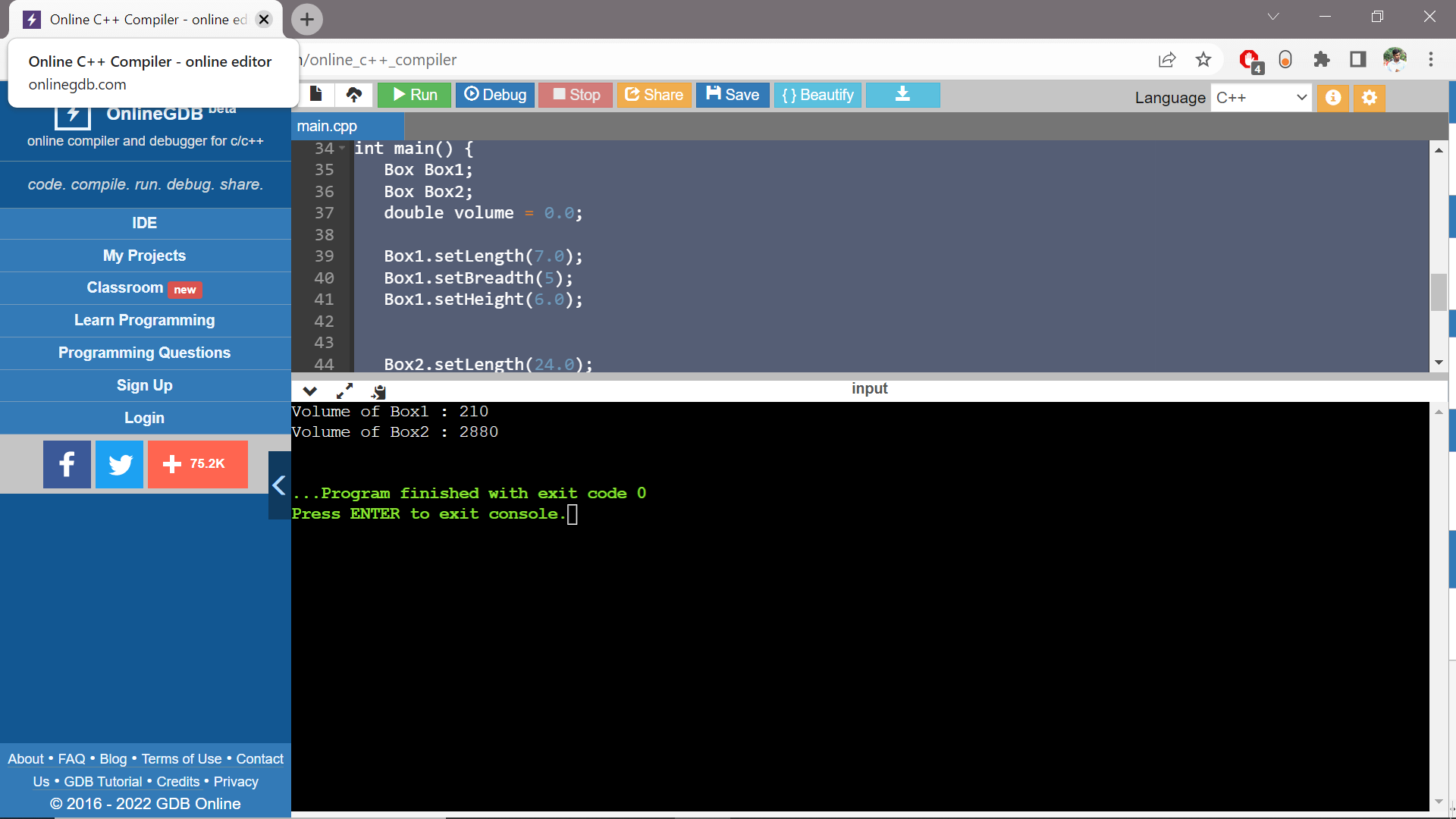
**volume = Box2.getVolume();**

**cout << "Volume of Box2 : " << volume <<endl;**

**return 0;**

**}**

**Output:**



A2. Write an example of a class having abstraction.

**Code:**

**#include <iostream>**

**using namespace std;**

**class impAbstraction**

**{**

**private:**

**int a, b;**

**public:**

**void set(int x, int y)**

**{**

**a = x;**

**b = y;**

**}**

**void displayOP()**

**{**

**cout<<"a = " <<a << endl;**

**cout<<"b = " << b << endl;**

**}**

**};**

**int main()**

**{**

**impAbstraction obj;**

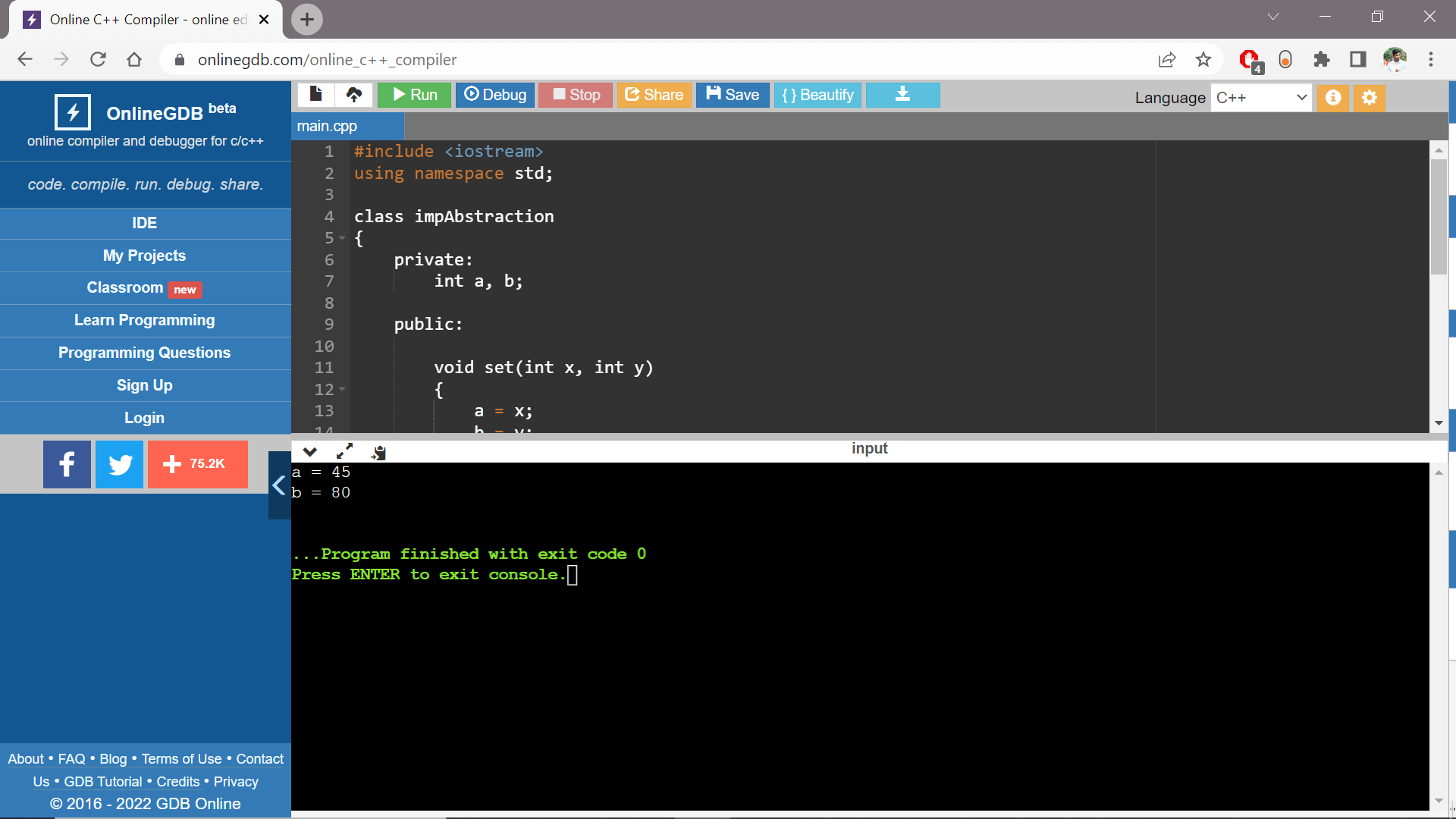
**obj.set(45, 80);**

**obj.displayOP();**

**return 0;**

**}**

Output:



A3.  Write an class having Encapsulation.

**Code:**

**#include <iostream>**

**using namespace std;**

**class Rectangle {**

**public:**

**int length;**

**int breadth;**

**Rectangle(int len, int bre) : length(len), breadth(bre) {}**

**int getArea() {**

**return length \* breadth;**

**}**

**};**

**int main() {**

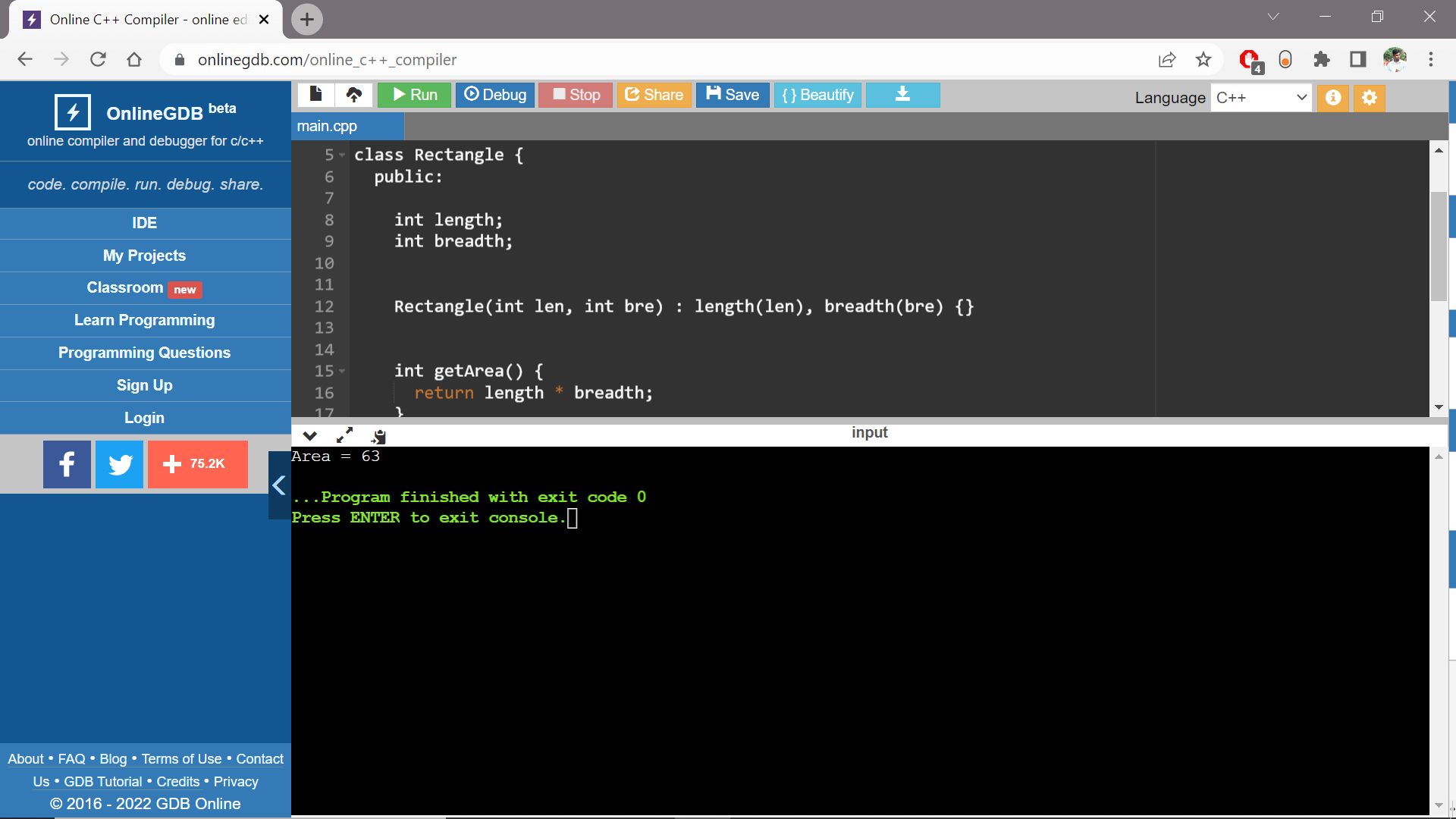
**Rectangle rect(9,7);**

**cout << "Area = " << rect.getArea();**

**return 0;**

**}**

**Output:**



A4.  Write an class having Polymorphisms.

**Code:**

**#include <iostream>**

**using namespace std;**

**class Shape {**

**protected:**

**int width, height;**

**public:**

**Shape( int a = 0, int b = 0){**

**width = a;**

**height = b;**

**}**

**virtual int area() {**

**cout << "Parent class area :" << width \* height << endl;**

**return width \* height;**

**}**

**};**

**class Rect: public Shape {**

**public:**

**Rect( int a = 0, int b = 0):Shape(a, b) { }**

**int area () {**

**cout << "Rectangle class area :" << width \* height << endl;**

**return (width \* height);**

**}**

**};**

**class Tri: public Shape {**

**public:**

**Tri( int a = 0, int b = 0):Shape(a, b) { }**

**int area () {**

**cout << "Triangle class area :" << (width \* height)/2 << endl;**

**return (width \* height / 2);**

**}**

**};**

**int main() {**

**Shape \*shape;**

**Rect rec(9,7);**

**Tri tri(8,5);**

**shape = &rec;**

**shape->area();**

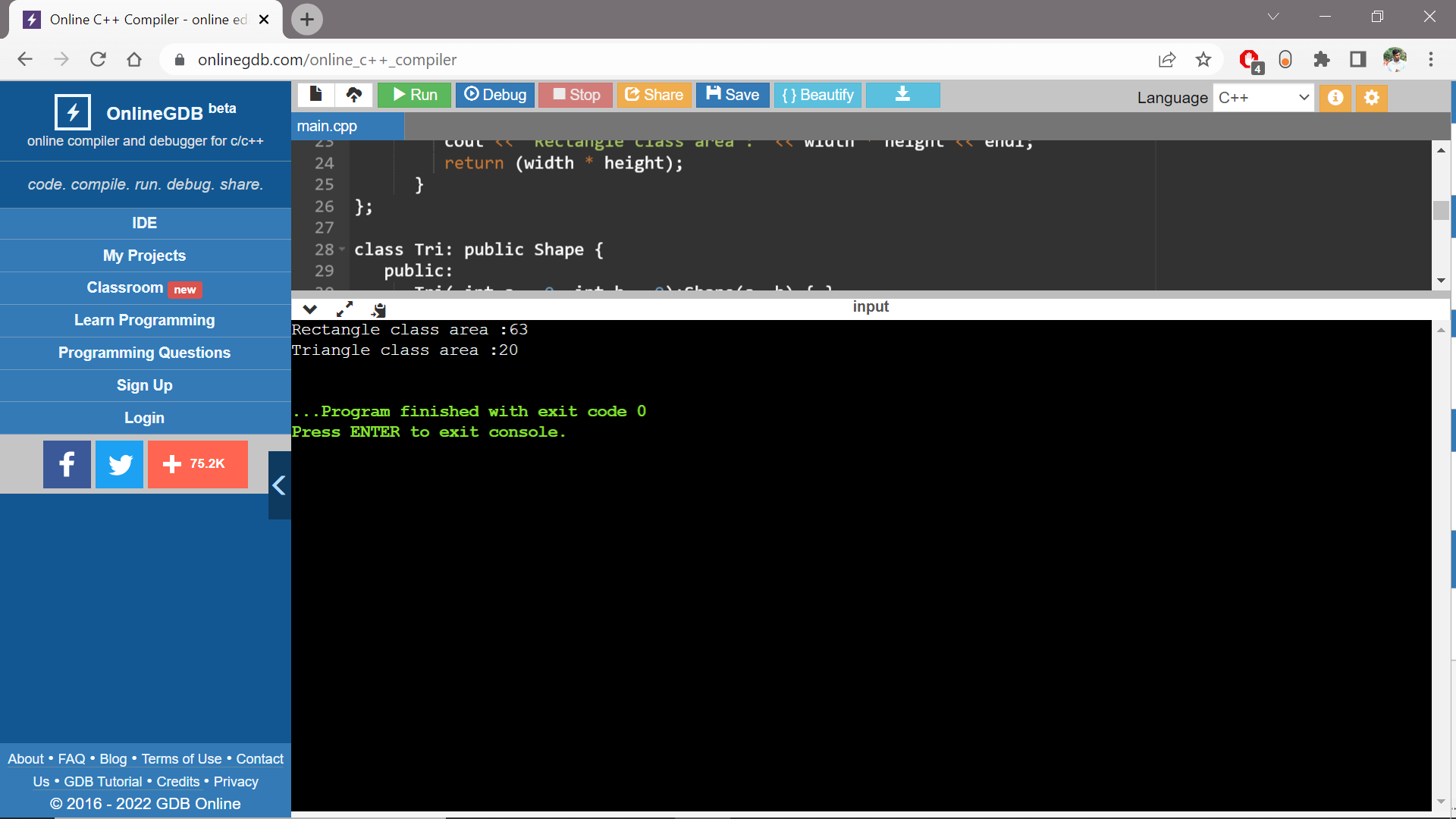
**shape = &tri;**

**shape->area();**

**return 0;**

**}**

**Output:**



A5.  Write an Class having different examples of Inheritance.

**Code:**

**Same code as above as it shows both inheritance and polymorphism properties.**

**Code:**

**#include <iostream>**

**using namespace std;**

**class Shape {**

**protected:**

**int width, height;**

**public:**

**Shape( int a = 0, int b = 0){**

**width = a;**

**height = b;**

**}**

**virtual int area() {**

**cout << "Parent class area :" << width \* height << endl;**

**return width \* height;**

**}**

**};**

**class Rect: public Shape {**

**public:**

**Rect( int a = 0, int b = 0):Shape(a, b) { }**

**int area () {**

**cout << "Rectangle class area :" << width \* height << endl;**

**return (width \* height);**

**}**

**};**

**class Tri: public Shape {**

**public:**

**Tri( int a = 0, int b = 0):Shape(a, b) { }**

**int area () {**

**cout << "Triangle class area :" << (width \* height)/2 << endl;**

**return (width \* height / 2);**

**}**

**};**

**int main() {**

**Shape \*shape;**

**Rect rec(9,7);**

**Tri tri(8,5);**

**shape = &rec;**

**shape->area();**

**shape = &tri;**

**shape->area();**

**return 0;**

**}**

**Output:**

Text

Description automatically generated

A6. Write an class BusBooking that has data members as ticketFare, nameOfPassanger, gender,age, destination, source and MemberFunction bookTicket(), cancelTicket(), calculateFare().

**Code:**

**#include <iostream>**

**using namespace std;**

**class BusBooking {**

**public:**

**double ticketFare =3000;**

**string nameofPassenger,source,destination;**

**char gender;**

**double hrs,total;**

**int age,noTicket;**

**void bookTicket(int,string,int,char,string,string);**

**double cancelTicket(double,double);**

**double calculateTicket(int,double);**

**};**

**void BusBooking:: bookTicket(int nT,string name,int agE,char gen,string sou,string dest){**

**nameofPassenger=name;**

**age=agE;**

**gender=gen;**

**noTicket=nT;**

**source=sou;**

**destination=dest;**

**cout<<"Name of the passenger: " << nameofPassenger << endl;**

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

**cout<<"Gender: "<<gender<< endl;**

**cout<<"The Number of Tickets booked: "<< noTicket << endl;**

**cout<<"Source of the Journey: " << source << endl;**

**cout<<"Destination of the Journey: " << destination << endl;**

**}**

**double BusBooking:: cancelTicket(double hours, double ttl){**

**hrs=hours;**

**total=ttl;**

**double loss=0;**

**if(hrs<=24.0){**

**cout<<"Full Amount Refunded : " << total <<endl;**

**}**

**else if(hrs>24.0 && hrs<=75.0){**

**loss=(60.0/100.0)\*total;**

**cout<<"Only 60 perecentage of your amount will be refunded." << endl;**

**total-=loss;**

**}**

**else{**

**cout << "Sorry you are not eligible for refund!" << endl;**

**total=0;**

**}**

**cout << "Total Amount refunded: " <<total;**

**return total;**

**}**

**double BusBooking:: calculateTicket(int nT,double ticketFare){**

**total=ticketFare\*nT;**

**cout << " The total ticket: " << total <<endl;**

**return total;**

**}**

**int main() {**

**BusBooking obj;**

**obj.bookTicket(20,"Saurav",22,'M',"Bangalore","New York");**

**obj.calculateTicket(20,3000);**

**obj.cancelTicket(90,obj.calculateTicket(20,3000));**

**return 0;**

**}**

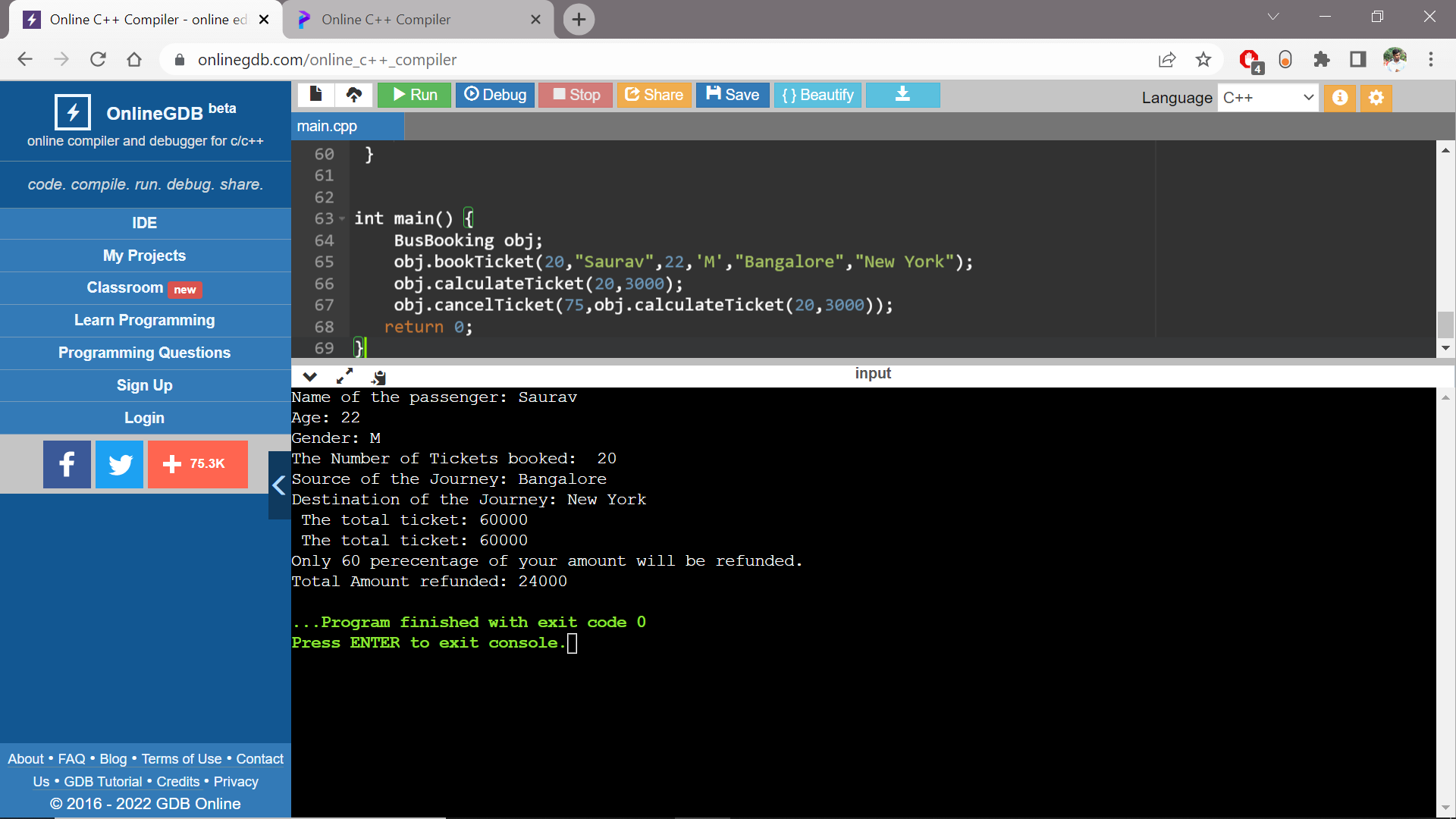
Output:

Case 1: If cancellation inside 24hrs of booking the ticket.

Text

Description automatically generated

**Case 2: If cancellation inside 75hrs of booking ticket.**



**Case 3: If cancellation is after 75 hrs of booking the ticket.**

