ASSIGNMENT inheritance

Que;WAP to implement a C++ program to find out the area of the rectangle and triangle using hierarchical inheritance .

Syntax-;

#include<iostream>

using namespace std;

class Area // base class

{ protected:

double length;

double height;

public:

Area(double l,double h) // parameterised constructor

{

length = l;

height = h;

}

};

class Rectangle : public Area // public inheritance

{

public:

Rectangle(double l,double h):Area(l,h){} // parameterised constructor

double Areaof()

{

return length\*height;

}

};

class Triangle : public Area

{

public:

Triangle(double l,double h):Area(l,h){}

double Areaof()

{

return 0.5\*length\*height;

}

};

int main()

{

Rectangle r(10,2);

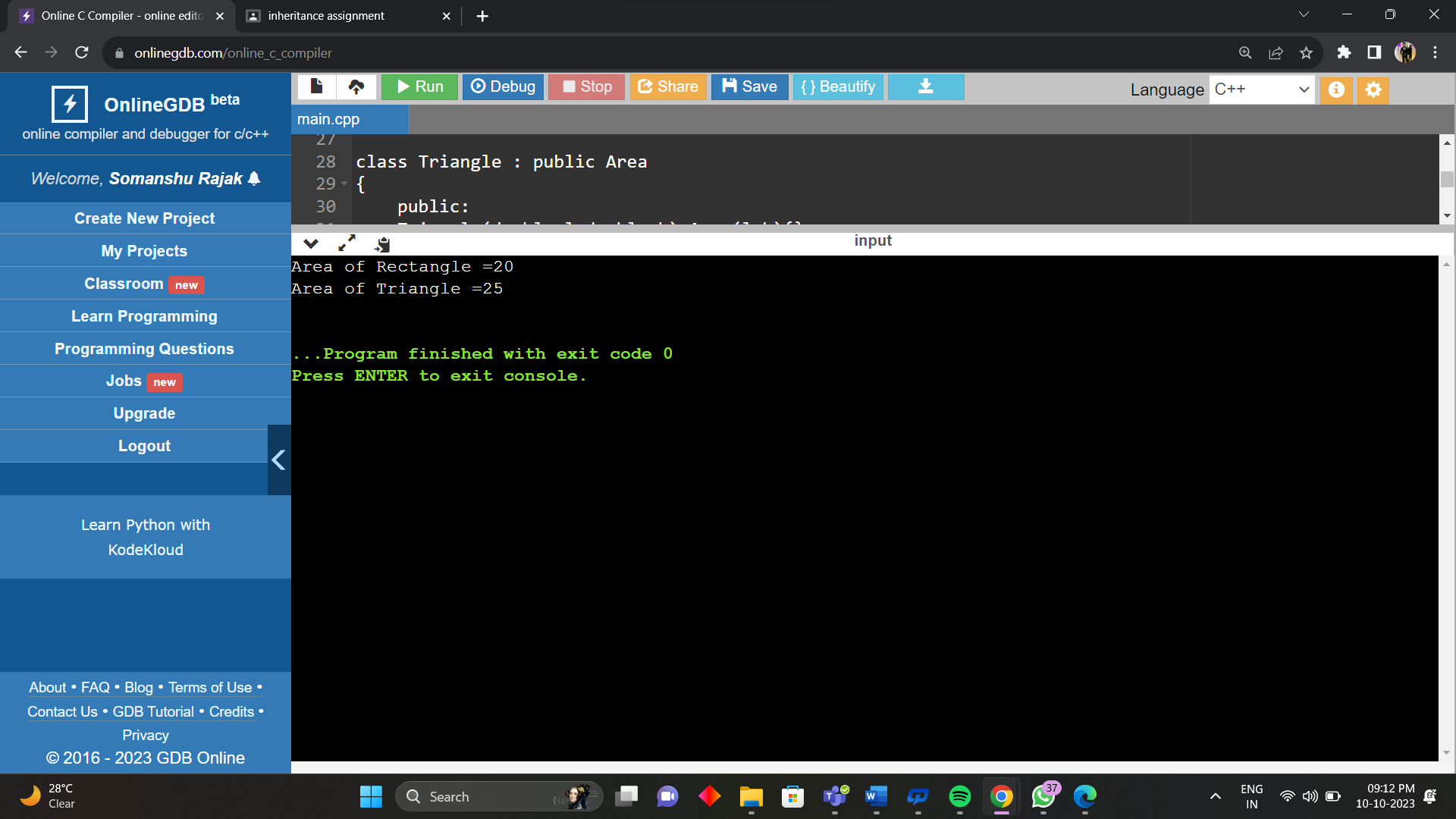
cout<<"Area of Rectangle ="<<r.Areaof()<<endl;

Triangle t(10,5);

cout<<"Are of Triangle ="<<t.Areaof()<<endl;

return 0;

}



Que ;WAP to implement a C++ program to find out the student details using multilevel inheritance.

#include<iostream>

using namespace std;

class Student

{

int roll;

char name[25];

public:

void input()

{

cout<<" Enter Student Name : ";

cin>>name;

cout<<" Enter Roll No. : ";

cin>>roll;

}

void display\_details()

{

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

cout<<" Roll No. : "<<roll<<endl;

}

};

class Exam : public Student //Class Exam derived from Class Student

{

public:

int sub1;

int sub2;

int sub3;

float per;

public:

void input()

{

Student::input();

cout<<"Enter Marks for Subject 1 : ";

cin>>sub1;

cout<<"Enter Marks for Subject 2 : ";

cin>>sub2;

cout<<"Enter Marks for Subject 3 : ";

cin>>sub3;

}

void display\_details()

{

Student::display\_details();

cout<<" Marks of Subject 1 : "<<sub1<<endl;

cout<<" Marks of Subject 2 : "<<sub2<<endl;

cout<<" Marks of Subject 3 : "<<sub3<<endl;

}

};

class Result : public Exam //Class Result derived from Class Exam

{

public:

void calculate ()

{

per = (sub1+sub2+sub3)/3.0;

cout<<" Total Percentage : "<<per;

cout<< endl;

}

};

int main()

{

Result s1;

s1.input();

s1.display\_details();

s1.calculate();

cout<<endl;

Result s2;

s2.input();

s2.display\_details();

s2.calculate();

cout<<endl;

Result s3;

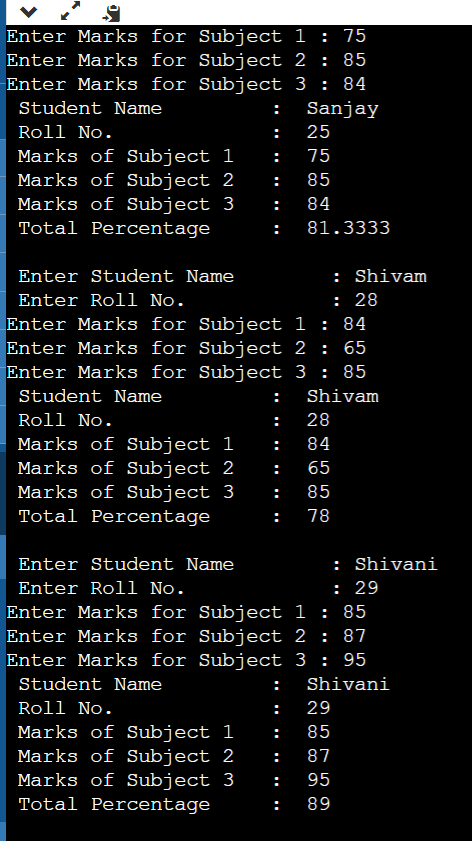
s3.input();

s3.display\_details();

s3.calculate();

return 0;

}



Que ;WAP to implement a C++ program to find out the student details and sport score using hybrid inheritance.

#include <iostream>

using namespace std;

class Student {

protected:

string name;

int roll\_no;

public:

Student( string n,int r) // Parameterized constructor

{

name = n;

roll\_no = r;

}

void display() // Function to display student details

{

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

cout << "Roll no: " << roll\_no << endl;

}

};

class Sport

{

protected:

int score;

public:

Sport(int s) // Parameterized constructor

{

score = s;

}

void display() // Function to display sport score

{

cout << "Sport score: " << score << endl;

}

};

class Result : public Student, public Sport

{

private:

int total;

public:

Result( string n,int r, int s) : Student(n,r),Sport(s) // Parameterized constructor

{

total = s;

}

void display() // Function to display result

{

Student::display();

Sport::display();

cout << "Total score: " << total << endl;

}

};

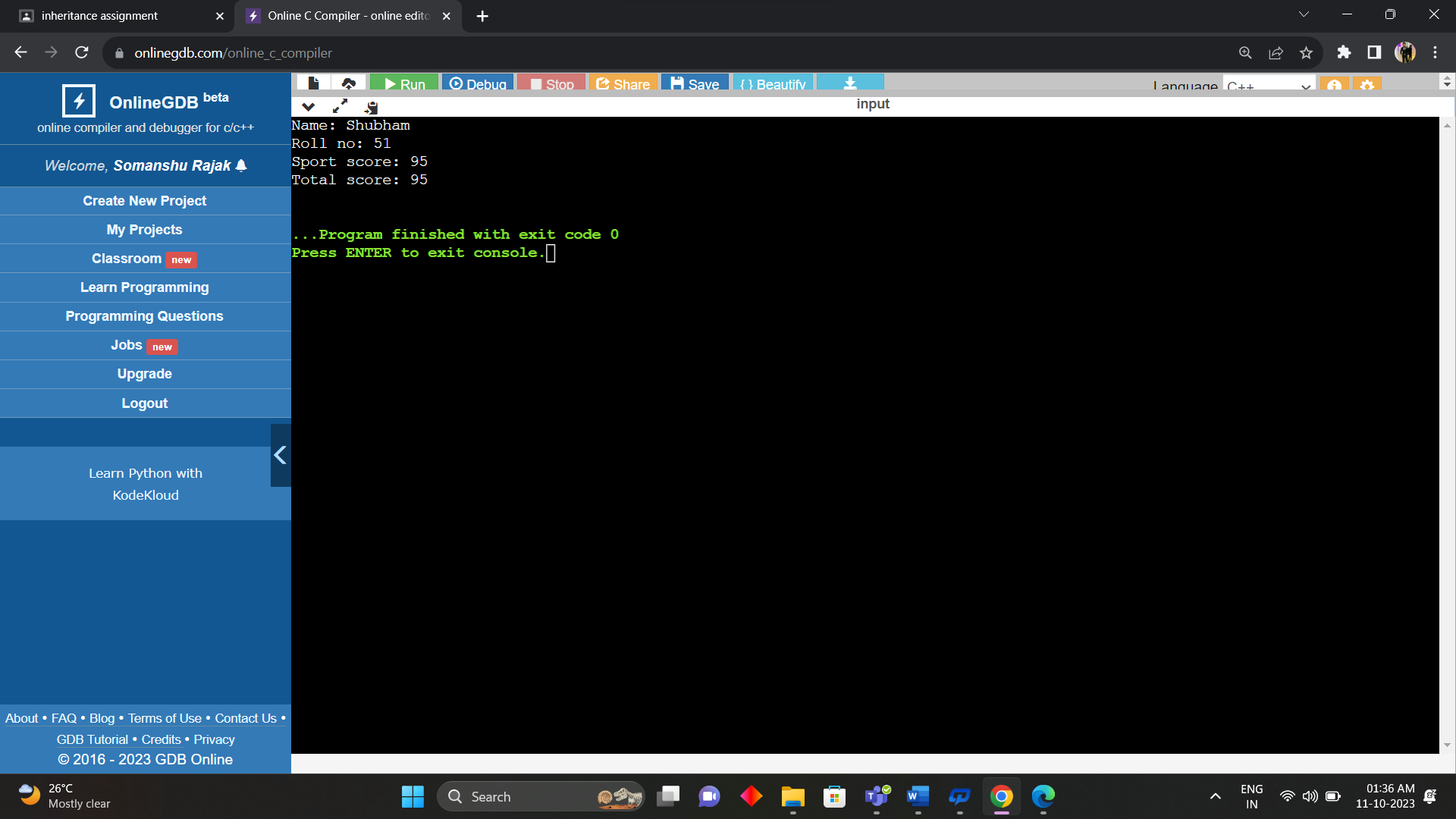
int main() {

Result R("Shubham" ,51, 95);

R.display();

return 0;

}



Que; Implement function overriding by creating class shape through which area of figures are calculated. Virtual function

#include <iostream>

#include <cmath>

using namespace std;

class Area

{

public:

virtual double calculate\_area()

{

return 0.0;

}

};

class Rectangle : public Area

{

private:

double length;

double width;

public:

Rectangle(double length, double width) : length(length), width(width) {}

double calculate\_area()

{

return length \* width;

}

};

class Circle : public Area

{

private:

double radius;

public:

Circle(double radius) : radius(radius) {}

double calculate\_area()

{

return 3.14 \* radius \* radius;

}

};

int main()

{

Rectangle Ar(4, 5);

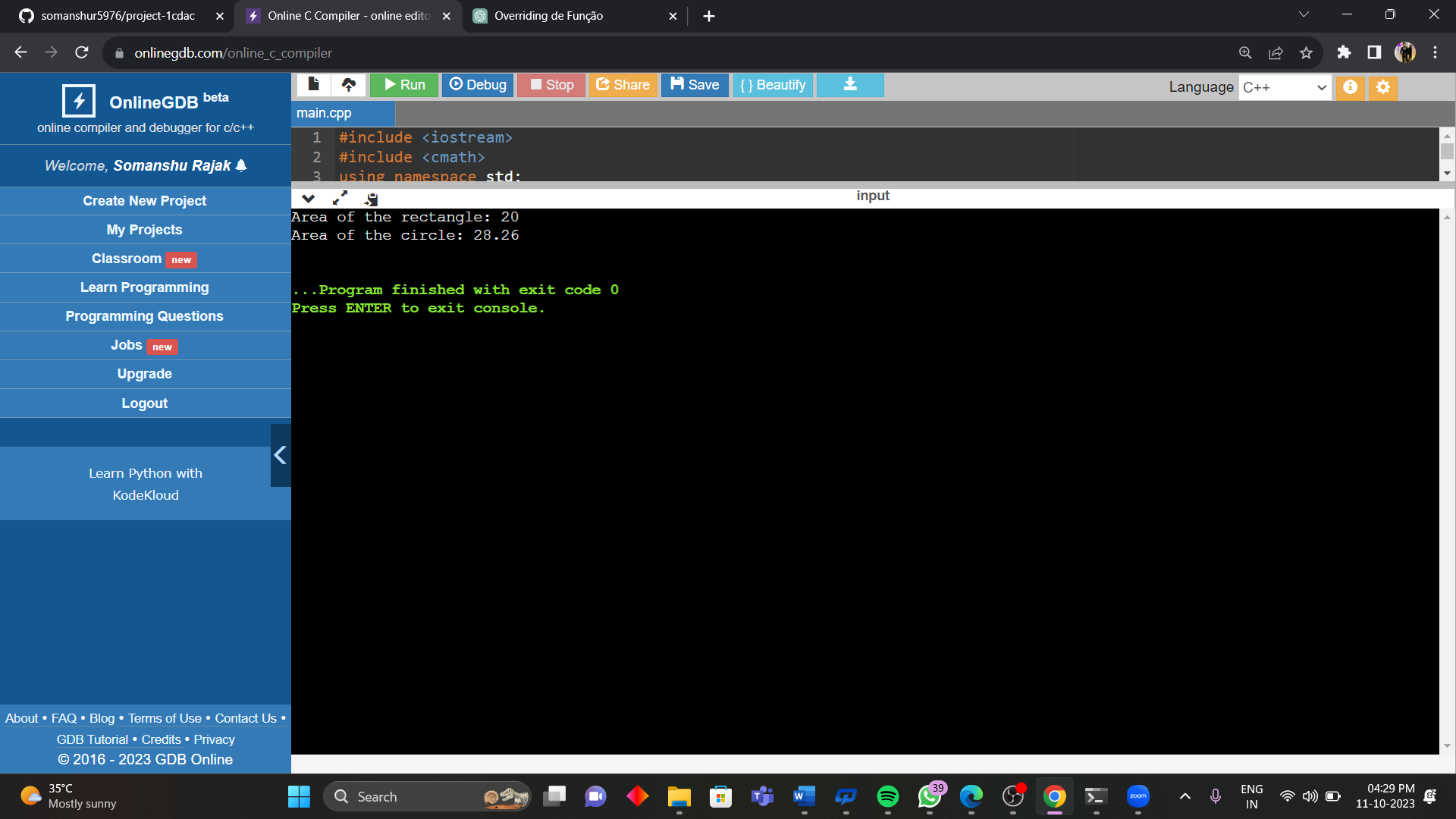
Circle Ac(3);

cout << "Area of the rectangle: " << Ar.calculate\_area() << endl;

cout << "Area of the circle: " << Ac.calculate\_area() << endl;

return 0;

}



Que; Class student contains roll number, name and course as data member and Input\_student and display\_student as member function. A derived class exam is created from the class student with publicly inherited. The derived class contains mark1, mark2, mark3 as marks of three subjects and input\_marks and display\_result as member function. Create an array of object of the exam class and display the result of 5 students. Try the same program with privately inheritance.

Publicly inherited

{

Result s1;

s1.input();

s1.display\_details();

s1.calculate();

cout<<endl;

Result s2;

s2.input();

s2.display\_details();

s2.calculate();

cout<<endl;

Result s3;

s3.input();

s3.display\_details();

s3.calculate();

cout<<endl;

Result s4;

s4.input();

s4.display\_details();

s4.calculate();

cout<<endl;

Result s5;

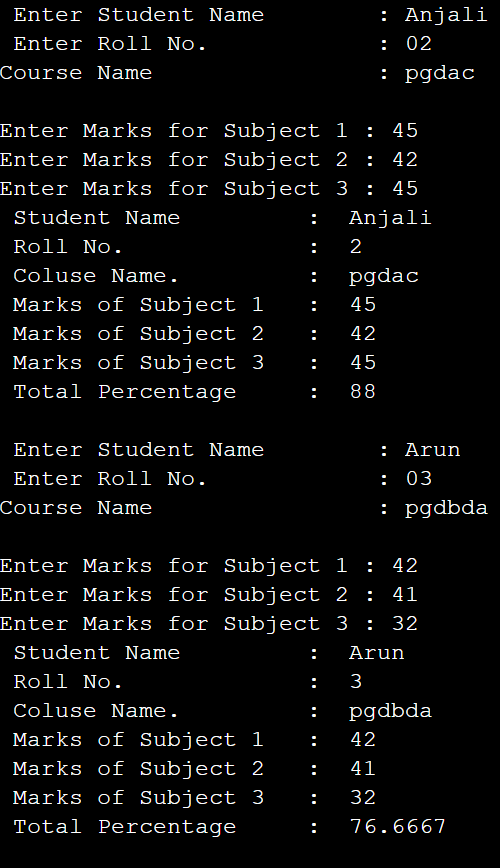
s5.input();

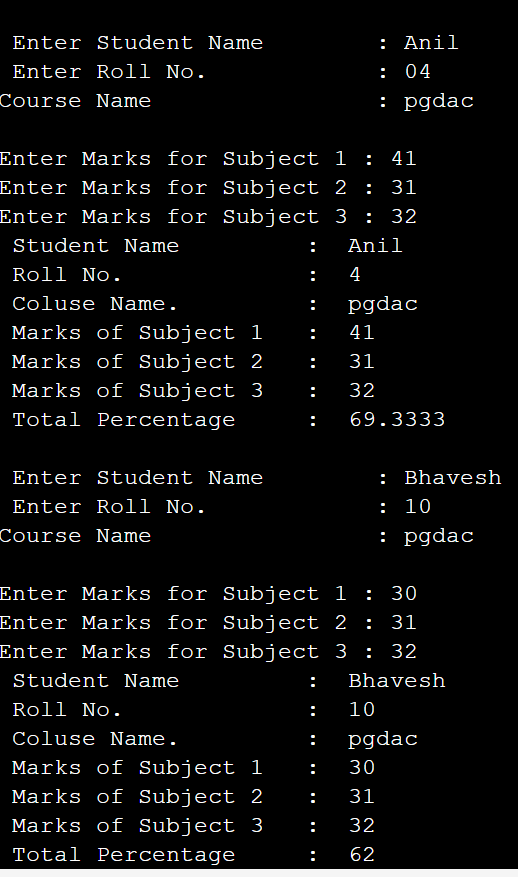
s5.display\_details();

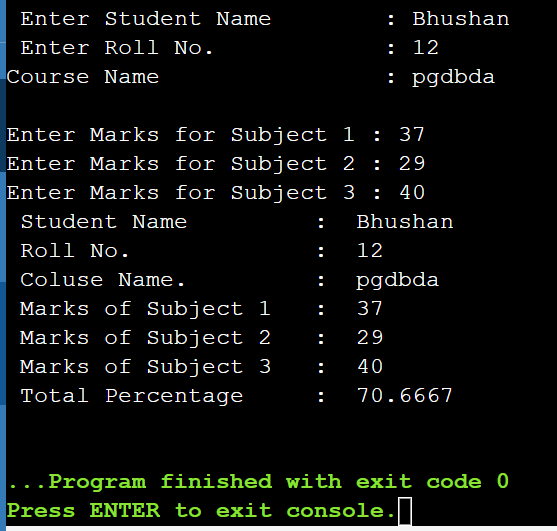
s5.calculate();

return 0;

}







Que; A University and a Company have jointly taken a project. Class University contains name of the university, department to which the project is assigned, person to whom the project is assigned. A function display is there to display the information. Class Company contains name of the company, Number of Engineers assigned, amount invested to do the project. A function display is there to display the information. Class Project is inherited from University and Company. It contains type of project, duration of project, amount granted to complete the project. A function display displays the related information. Write a C++ program to implement this and display all information except amount invested by company from Project class

#include<iostream>

using namespace std;

class University

{

private:

string UniversityName;

string deptName;

string personName;

public:

University()

{

cout<<"Enter the name of university: ";

cin>>UniversityName;

cout<<"ENter the name of the Department: ";

cin>>deptName;

cout<<"Enter the Person name: ";

cin>>personName;

}

void DisplayData()

{

cout<<"Name of the University: "<<UniversityName<<endl;

cout<<"Department Name: "<<deptName<<endl;

cout<<"Person Name: "<<personName<<endl;

}

};

class Company

{

private:

string company\_name;

int No\_of\_engineer;

double Amount;

public:

Company()

{

cout<<"Enter the Company name: ";

cin>>company\_name;

cout<<"Enter the No Of Engineer: ";

cin>>No\_of\_engineer;

cout<<"Enter the Amount invested to the project in usd: ";

cin>>Amount;

}

void DisplayData()

{

cout<<"Name of the Company : "<<company\_name<<endl;

cout<<"No of Engineer: "<<No\_of\_engineer;

cout<<"Amount: "<<Amount<<" "<<"$"<<endl;

}

};

class Project :public University , public Company

{

private:

string ProjectType;

int ProjectDuration;

double ProjectAmount;

public:

Project()

{

cout<<"Enter the type of project: ";

cin>>ProjectType;

cout<<"Enter the Project Duration: ";

cin>>ProjectDuration;

cout<<"Enter the Project Amount in usd: ";

cin>>ProjectAmount;

cout<<endl;

}

void DisplayData()

{

cout<<"Type of project: "<<ProjectType<<endl;

cout<<"Duration of the Project: "<<ProjectDuration<<endl;

cout<<"Total Amount of the project: "<<ProjectAmount<<" "<<"$"<<endl;

}

};

int main()

{

Project project1;

project1.University::DisplayData();

project1.Company::DisplayData();

project1.DisplayData();

return 0;

}

