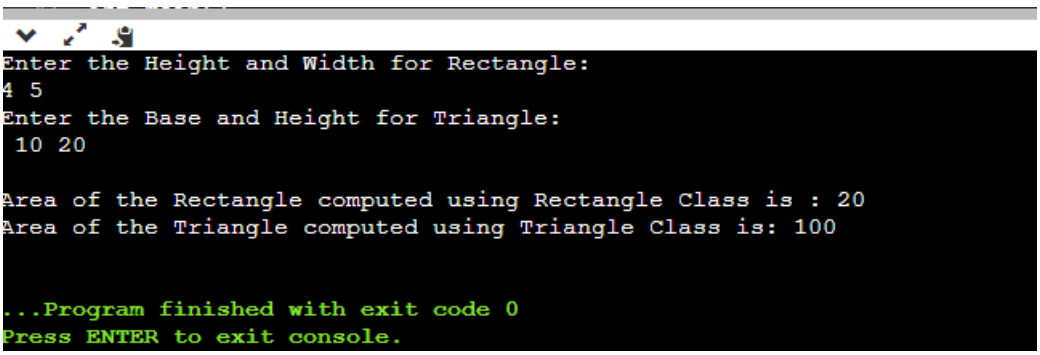
ASSIGNMENT

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

Code:

```
#include <iostream>
using namespace std;

// defining the class Shape to demonstrate the concept of
Hierarchial Inheritence in CPP
class Shape
{
protected:
float width, height;
// public members are accessible everywhere
public:
void setDimensions(float w, float h)
{
width = w;
height = h;
}
};

// Class Rectangle inherites the Shape class
class Rectangle : public Shape
{
// Method Overriding
public:
float area()
{
return (width * height);
}
};

// Class Triangle inherites the Shape class
class Triangle : public Shape
{
// Method Overriding
public:
float area()
{
return (width * height / 2);
```

```
}
};

// Defining the main method to access the members of the class
int main()
{
int rHeight,width,tHeight,base;
cout << "Enter the Height and Width for Rectangle: \n";
cin >>rHeight>>width;
cout << "Enter the Base and Height for Triangle: \n";
cin >>base>>tHeight;

// Declaring the Class objects to access the class members
Rectangle rectangle;
Triangle triangle;

rectangle.setDimensions(rHeight, width);
triangle.setDimensions(base, tHeight);

cout << "\nArea of the Rectangle computed using Rectangle Class
is : " << rectangle.area() << "\n";
cout << "Area of the Triangle computed using Triangle Class is: "
<< triangle.area() << endl;

return 0;
}
```



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

Code:

```
#include<iostream>
using namespace std;

class Student
{
int roll;
char name[25];
public:
void getdata()
{
cout<<"\n ----------------------------------------";
cout<<"\n Enter Roll No.           : ";
cin>>roll;
cout<<"\n Enter Student Name       : ";
cin>>name;
}
void putdata()
{
cout<<"\n ----------------------------------------";
cout<<"\n ********** Student Marklist **********";
cout<<"\n ----------------------------------------";
cout<<"\n Roll No.            :  "<<roll;
cout<<"\n Student Name        :  "<<name<<endl;
}
};
class StudentExam : public Student      //Class StudentExam
derived from Class Student
{
public:
int sub1, sub2, sub3, sub4, sub5, sub6;
float per;
public:
void accept_data()
{
getdata();
cout<<"\n Enter Marks for Subject 1 : ";
cin>>sub1;
cout<<"\n Enter Marks for Subject 2 : ";
cin>>sub2;
```

```
cout<<"\n Enter Marks for Subject 3 : ";
cin>>sub3;
cout<<"\n Enter Marks for Subject 4 : ";
cin>>sub4;
cout<<"\n Enter Marks for Subject 5 : ";
cin>>sub5;
cout<<"\n Enter Marks for Subject 6 : ";
cin>>sub6;
}
void display_data()
{
putdata();
cout<<"\n Marks of Subject 1   :  "<<sub1;
cout<<"\n Marks of Subject 2   :  "<<sub2;
cout<<"\n Marks of Subject 3   :  "<<sub3;
cout<<"\n Marks of Subject 4   :  "<<sub4;
cout<<"\n Marks of Subject 5   :  "<<sub5;
cout<<"\n Marks of Subject 6   :  "<<sub6;
}
};
class StudentResult : public StudentExam      //Class StudentResult
derived from Class StudentExam
{
public:
void calculate ()
{
per = (sub1+sub2+sub3+sub4+sub5+sub6)/6.0;
cout<<"\n\n Total Percentage     :  "<<per;
cout<<"\n ------------------------------------- \n";
}
};
int main()
{
StudentResult str;     //Object 'str' is created of derived Class
StudentResult
int cnt, i;
cout<<"\n Enter No. of Students You Want? : ";
cin>>cnt;
for(i=0; i<cnt; i++)
{
str.accept_data();
str.display_data();
str.calculate();
```

```
}
return 0;
}
```

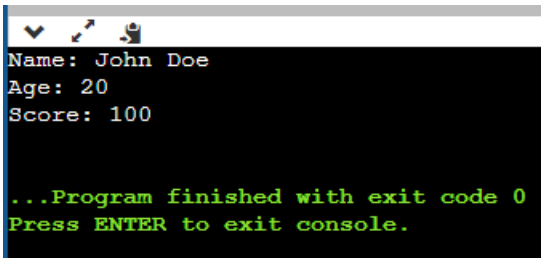
```
Enter No. of Students You Want? : 1

----------------------------------------
Enter Roll No.             : 10

Enter Student Name         : Rohit

Enter Marks for Subject 1 : 40

Enter Marks for Subject 2 : 50

Enter Marks for Subject 3 : 60

Enter Marks for Subject 4 : 60

Enter Marks for Subject 5 : 70

Enter Marks for Subject 6 : 80

----------------------------------------
********** Student Marklist **********
----------------------------------------
Roll No.              :  10
Student Name          :  Rohit

Marks of Subject 1    :  40
Marks of Subject 2    :  50
Marks of Subject 3    :  60
Marks of Subject 4    :  60
Marks of Subject 5    :  70
Marks of Subject 6    :  80

Total Percentage      :  60
----------------------------------------


...Program finished with exit code 0
Press ENTER to exit console.
```

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

Code:

```
#include <iostream>

// Base class
class Student {
public:
std::string name;
int age;

void printDetails() {
std::cout << "Name: " << name << "\nAge: " << age << "\n";
}
};

// Derived class
class Athlete : public Student {
public:
int score;

void printScore() {
std::cout << "Score: " << score << "\n";
}
};

// Main function
int main() {
// Create an object of the Athlete class
Athlete athlete;

// Set the student details
athlete.name = "John Doe";
athlete.age = 20;

// Set the athlete's score
athlete.score = 100;

// Print the student details
athlete.printDetails();
```

```
// Print the athlete's score
athlete.printScore();

return 0;
}
```

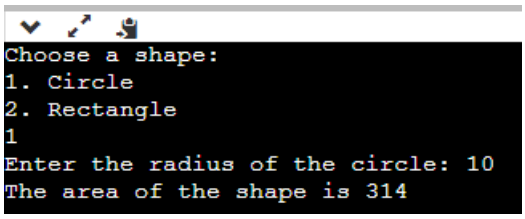


Implement function overriding by creating class shape through which area of figures are calculated.

Code:

```
#include <iostream>

using namespace std;

class Shape {
public:
virtual double getArea() = 0;
};

class Circle : public Shape {
public:
double getArea()  {
double radius;
cout << "Enter the radius of the circle: ";
cin >> radius;
return 3.14 * radius * radius;
}
};

class Rectangle : public Shape {
public:
double getArea()  {
double length, width;
cout << "Enter the length and width of the rectangle: ";
cin >> length >> width;
return length * width;
```

```
}
};

int main() {
Shape* shape;

cout << "Choose a shape: \n1. Circle \n2. Rectangle \n";
int choice;
cin >> choice;

if (choice == 1) {
shape = new Circle();
} else {
shape = new Rectangle();
}

double area = shape->getArea();
cout << "The area of the shape is " << area << endl;

return 0;
}
```



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.

Code:

```
#include<iostream>
using namespace std;

class student {
protected:
int roll_number;
string name;
string course;

public:
void input_student() {
cout << "Enter roll number: ";
cin >> roll_number;
cout << "Enter name: ";
cin >> name;
cout << "Enter course: ";
cin >> course;
}

void display_student() {
cout << "Roll number: " << roll_number << endl;
cout << "Name: " << name << endl;
cout << "Course: " << course << endl;
}
};

class exam : public student {
private:
int mark1, mark2, mark3;

public:
void input_marks() {
```

```
cout << "Enter marks for subject 1: ";
cin >> mark1;
cout << "Enter marks for subject 2: ";
cin >> mark2;
cout << "Enter marks for subject 3: ";
cin >> mark3;
}

void display_result() {
cout << "Marks for subject 1: " << mark1 << endl;
cout << "Marks for subject 2: " << mark2 << endl;
cout << "Marks for subject 3: " << mark3 << endl;
}
};

int main() {
exam students[5];

for(int i=0; i<5; i++) {
cout << "Enter details for student " << i+1 << ":" << endl;
students[i].input_student();
students[i].input_marks();
}

cout << endl;

for(int i=0; i<5; i++) {
cout << "Details for student " << i+1 << ":" << endl;
students[i].display_student();
students[i].display_result();
cout << endl;
}

return 0;
}
```

```
Details for student 1:
Roll number: 10
Name: Rohit
Course: maths
Marks for subject 1: 50
Marks for subject 2: 60
Marks for subject 3: 70

Details for student 2:
Roll number: 11
Name: Karan
Course: English
Marks for subject 1: 50
Marks for subject 2: 60
Marks for subject 3: 80

Details for student 3:
Roll number: 13
Name: Himanshu
Course: hindi
Marks for subject 1: 50
Marks for subject 2: 40
Marks for subject 3: 90

Details for student 4:
Roll number: 5
Name: yugank
Course: science
Marks for subject 1: 50
Marks for subject 2: 60
Marks for subject 3: 75

Details for student 5:
Roll number: 16
Name: naman
Course: Biology
Marks for subject 1: 60
```

```
Details for student 1:
Roll number: 10
Name: Rohit
Course: maths
Marks for subject 1: 50
Marks for subject 2: 60
Marks for subject 3: 70

Details for student 2:
Roll number: 11
Name: Karan
Course: English
Marks for subject 1: 50
Marks for subject 2: 60
Marks for subject 3: 80

Details for student 3:
Roll number: 13
Name: Himanshu
Course: hindi
Marks for subject 1: 50
Marks for subject 2: 40
Marks for subject 3: 90

Details for student 4:
Roll number: 5
Name: yugank
Course: science
Marks for subject 1: 50
Marks for subject 2: 60
Marks for subject 3: 75

Details for student 5:
Roll number: 16
Name: naman
Course: Biology
Marks for subject 1: 60
```

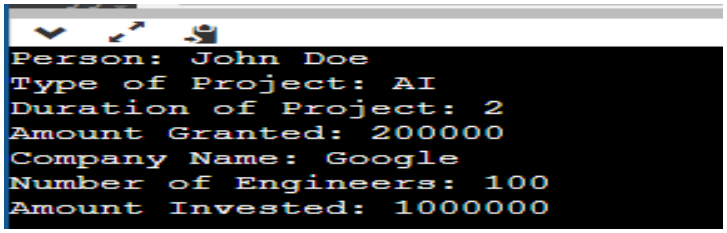
A University and a Company have jointly taken a project. Class University contains name of the university, department to which the project is assigned, person to whom the project is assigned. A function display is there to display the information. Class Company contains name of the company, Number of 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

Code:

```
#include <iostream>
#include <string>

using namespace std;

class University {
protected:
string name;
string department;
string person;

public:
University(string name, string department, string person)
: name(name), department(department), person(person) {}

void display() {
cout << "University Name: " << name << endl;
cout << "Department: " << department << endl;
cout << "Person: " << person << endl;
}
};

class Company {
protected:
string name;
int numberOfEngineers;
int amountInvested;

public:
```

```
Company(string name, int numberOfEngineers, int
amountInvested)
: name(name), numberOfEngineers(numberOfEngineers),
amountInvested(amountInvested) {}

void display() {
cout << "Company Name: " << name << endl;
cout << "Number of Engineers: " << numberOfEngineers << endl;
cout << "Amount Invested: " << amountInvested << endl;
}
};

class Project : public University, public Company {
private:
string typeOfProject;
int durationOfProject;
int amountGranted;

public:
Project(string name, string department, string person, string
nameOfCompany, int numberOfEngineers, int amountInvested,
string typeOfProject, int durationOfProject, int amountGranted)
: University(name, department, person),
Company(nameOfCompany, numberOfEngineers,
amountInvested), typeOfProject(typeOfProject),
durationOfProject(durationOfProject),
amountGranted(amountGranted) {}

void display() {
University::display();
cout << "Type of Project: " << typeOfProject << endl;
cout << "Duration of Project: " << durationOfProject << endl;
cout << "Amount Granted: " << amountGranted << endl;
Company::display();
}
};

int main() {
Project project("Harvard University", "Computer Science", "John
Doe", "Google", 100, 1000000, "AI", 2, 200000);
project.display();

return 0;
```

}


Person: John Doe
Type of Project: AI
Duration of Project: 2
Amount Granted: 200000
Company Name: Google
Number of Engineers: 100
Amount Invested: 1000000