**ASSIGNMENT**

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

**Solution :**

#include <iostream>

using namespace std;

class area {

protected:

int x;

int y;

public:

double a(int ab){

return ab;

}

};

class triangle : public area {

public:

triangle(int base, int height) {

x = base;

y = height;

}

double a() {

return 0.5 \* x \* y;

}

};

class rectangle : public area {

public:

rectangle(int length, int width) {

x = length;

y = width;

}

double a() {

return x \* y;

}

};

int main() {

int base, height,length,width;

cout << "Enter the base and height of the triangle: ";

cin >> base >> height;

triangle t(base, height);

cout << "Area of the triangle: " << t.a() << endl;

cout << "Enter the length and width of the rectangle: ";

cin >> length >> width;

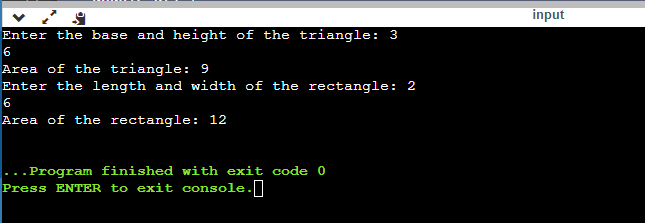
rectangle r(length, width);

cout << "Area of the rectangle: " << r.a() <<endl;

return 0;

}

**Result :**

****

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

**Solution:**

#include <iostream>

#include <string>

using namespace std;

class Student {

protected:

string name;

int rollNumber;

public:

void inputStudent() {

cout << "Enter Student Name: ";

cin.ignore();

getline(cin, name);

cout << "Enter Roll Number: ";

cin >> rollNumber;

}

void displayStudent() {

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

cout << "Roll Number: " << rollNumber << endl;

}

};

class Exam : public Student {

protected:

int marks[3];

public:

void inputExamMarks() {

inputStudent();

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

for (int i = 0; i < 3; i++) {

cin >> marks[i];

}

}

void displayExamResult() {

displayStudent();

cout << "Exam Results:" << endl;

for (int i = 0; i < 3; i++) {

cout << "Subject " << i + 1 << " Marks: " << marks[i] << endl;

}

}

};

class Result : public Exam {

public:

void displayResult() {

displayExamResult();

int totalMarks = 0;

for (int i = 0; i < 3; i++) {

totalMarks += marks[i];

}

float average = totalMarks/ 3;

cout << "Total Marks: " << totalMarks << endl;

cout << "Average Marks: " << average << endl;

}

};

int main() {

Result studentResult;

cout << "Enter Student Exam Details:" << endl;

studentResult.inputExamMarks();

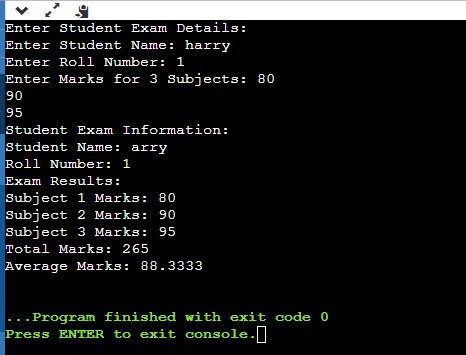
cout << "Student Exam Information:" << endl;

studentResult.displayResult();

return 0;

}

Result :



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

**Solution :**

#include<iostream>

using namespace std;

class student {

int id;

char name[30];

public:

void getstu() {

cout << "Enter roll no and name: ";

cin >> id >> name;

}

};

class marks : public student {

protected:

int a, b, c;

public:

void getmarks() {

cout << "Enter 3 subject marks: ";

cin >> a >> b >> c;

}

};

class sports {

protected:

int sp;

public:

void getsports() {

cout << "Enter sports marks: ";

cin >> sp;

}

};

class result : public marks, public sports {

public:

void show() {

double tot = a + b + c;

float avg = tot/ 3.0;

cout << "Total = " << tot << endl;

cout << "Average = " << avg << endl;

cout << "Average + Sports marks = " << avg + sp << endl;

}

};

int main() {

result r;

r.getstu();

r.getmarks();

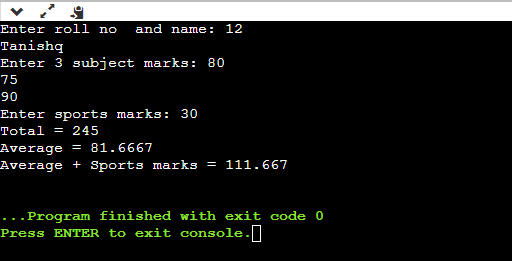
r.getsports();

r.show();

return 0;

}

**Result:**



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

**Solution :**

#include <iostream>

using namespace std;

class shape {

protected:

int x;

int y;

public:

virtual double area() = 0;

};

class triangle : public shape {

public:

triangle(int base, int height) {

x = base;

y = height;

}

double area() {

return 0.5 \* x \* y;

}

};

class rectangle : public shape {

public:

rectangle(int length, int width) {

x = length;

y = width;

}

double area() {

return x \* y;

}

};

class square : public shape {

public:

square(int side) {

x = side;

}

double area() {

return x \* x;

}

};

int main() {

int base, height, length, width, side;

cout << "Enter the base and height of the triangle: ";

cin >> base >> height;

triangle t(base, height);

cout << "Area of the triangle: " << t.area() << endl;

cout << "Enter the side of the square: ";

cin >> side;

square s(side);

cout << "Area of the square: " << s.area() << endl;

cout << "Enter the length and width of the rectangle: ";

cin >> length >> width;

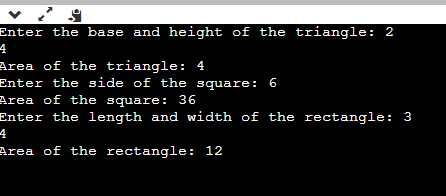
rectangle r(length, width);

cout << "Area of the rectangle: " << r.area() << endl;

return 0;

}

**Output:**

****

**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 anddisplay the result of 5 students. Try the same program with privately inheritance.**

**Solution :**

#include <iostream>

#include <string>

using namespace std;

class student {

protected:

int rollno;

string name;

string course;

public:

void input() {

cout << "Enter Name: ";

cin.ignore();

getline(cin, name);

cout << "Enter Roll Number: ";

cin >> rollno;

cin.ignore();

cout << "Enter Course: ";

cin.ignore();

getline(cin, course);

}

void display\_student() {

cout << "Roll Number: " << rollno << endl;

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

cout << "Course: " << course << endl;

}

};

class exam : public student {

protected:

int mark1, mark2, mark3;

public:

void marks() {

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

cin >> mark1 >> mark2 >> mark3;

}

void display\_result() {

display\_student();

cout << "Marks in Subject 1: " << mark1 << endl;

cout << "Marks in Subject 2: " << mark2 << endl;

cout << "Marks in 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();

students[i].marks();

}

cout << "Results for 5 students:" << endl;

for (int i = 0; i < 5; i++) {

cout << "student " << i + 1 << ":" << endl;

students[i].display\_result();

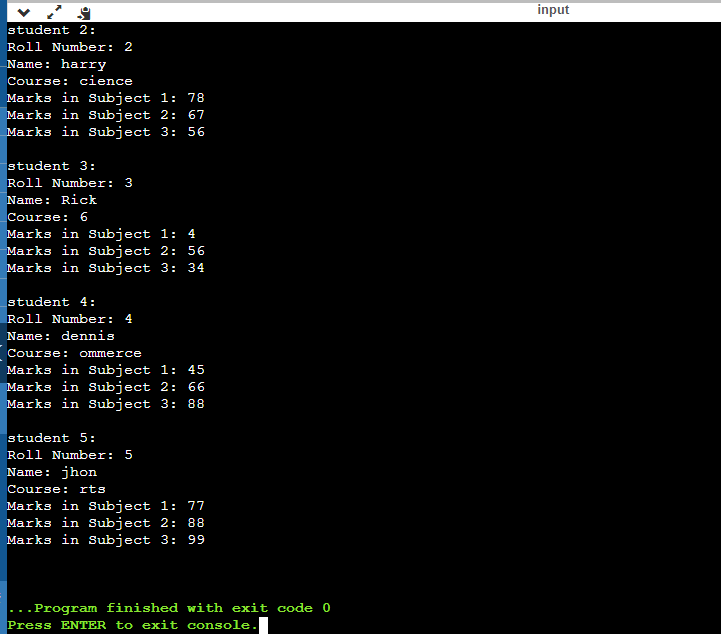
cout << endl;

}

return 0;

}

Result :



**B) while using private inheritance, which makes the base class's members private in the derived class. As a result, you can't directly access the base class's members from the derived class or from objects of the derived class.**

**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**

**Solution ;**

#include <iostream>

#include <string>

using namespace std;

class university {

protected:

string name;

string department;

string person;

public:

void inputUniversityInfo() {

cout << "Enter University Name: ";

cin >> name;

cout << "Enter Department: ";

cin >> department;

cout << "Enter Assigned Person: ";

cin >> person;

}

void displayUniversityInfo() {

cout << "University: " << name << endl;

cout << "Department: " << department << endl;

cout << "Assigned Person: " << person << endl;

}

};

class company {

protected:

string companyname;

int engineers\_assigned;

float amountinvested;

public:

void inputCompanyInfo() {

cout << "Enter Company Name: ";

cin >> companyname;

cout << "Enter Number of Engineers Assigned: ";

cin >> engineers\_assigned;

cout << "Enter Amount Invested: ";

cin >> amountinvested;

}

void displayCompanyInfo() {

cout << "Company: " << companyname << endl;

cout << "Engineers Assigned: " << engineers\_assigned << endl;

cout << "Amount Invested: " << amountinvested << endl;

}

};

class project : public university, public company {

protected:

string projectType;

int projectDuration;

float amountGranted;

public:

void inputProjectInfo() {

inputUniversityInfo();

inputCompanyInfo();

cout << "Enter Project Type: ";

cin >> projectType;

cout << "Enter Project Duration (in months): ";

cin >> projectDuration;

cout << "Enter Amount Granted: ";

cin >> amountGranted;

}

void displayProjectInfo() {

displayUniversityInfo();

displayCompanyInfo();

cout << "Project Type: " << projectType << endl;

cout << "Project Duration: " << projectDuration << " months" << endl;

cout << "Amount Granted: " << amountGranted << endl;

}

};

int main() {

project project;

cout << "Enter Project Information:" << endl;

project.inputProjectInfo();

cout << "Project Information (Excluding Investment):" << endl;

project.displayProjectInfo();

return 0;

}

**Output:**

