**Assignment-4**

**Inheritance: Derived Class declaration, Public, Private and Protected Inheritance, Forms of inheritance**

**1. Most of the time we use public mode of inheritance, for example class Derived:**

**public Base{} Try protected and private access modifiers to understand the difference**

**of various modes of inheritance.**

#include<iostream>

using namespace std;

class base{

private:

int priv\_i;

protected:

int pro\_i;

public:

int pub\_i;

base(){

priv\_i=10;

pro\_i=20;

pub\_i=30;

}

};

class derived1:public base{

public:

void getval(){

cout<<"Pro\_i in derived1 is: "<<pro\_i<<endl;

}

};

class derived2:protected base{

public:

void getval(){

cout<<"Pro\_i in derived2 is: "<<pro\_i<<endl;

cout<<"Pub\_i in derived2 is: "<<pub\_i<<endl;

}

};

class derived3:private base{

public:

void getval(){

cout<<"Pro\_i in derived3 is: "<<pro\_i<<endl;

cout<<"Pub\_i in derived3 is: "<<pub\_i<<endl;

}

};

int main(){

derived1 obj1;

cout<<"In derived1, pub\_i is: "<<obj1.pub\_i<<endl;

//public data member of base is inerited as public in derived1

obj1.getval();

//protected and public of base can only be accessed by member function of derived1

derived2 obj2;

obj2.getval();

derived3 obj3;

obj3.getval();

//protected and public of base can only be accessed by member function of derived1

//private members are not inherited irrespective of the access modifier

return 0;

}

**2. Write a C++ program to demonstrate example of hierarchical inheritance to get square**

**and cube of a number.**

#include<iostream>

using namespace std;

class number{

protected:

int num;

public:

void getnum(){

cout<<"Enter num: ";

cin>>num;

}

};

class square: public number{

public:

void getsq(){

cout<<"Square is: "<<num\*num;

}

};

class cube: public number{

public:

void getcu(){

cout<<"Cube is: "<<num\*num\*num;

}

};

int main(){

int ans;

cout<<"Enter 1 to square and 2 to cube"<<endl;

cin>>ans;

switch(ans){

case 1: {

square obj1;

obj1.getnum();

obj1.getsq();

break;

}

case 2: {

cube obj2;

obj2.getnum();

obj2.getcu();

break;

}

default: cout<<"Invalid";

}

return 0;

}

**3. Make a class named Fruit with a data member to calculate the number of fruits in a**

**basket. Create two other class named Apples and Mangoes to calculate the number of**

**apples and mangoes in the basket. Print the number of fruits of each type and the total**

**number of fruits in the basket.**

#include<iostream>

using namespace std;

class mango{

protected:

int mango\_no;

public:

void getmango(){

cout<<"Enter number of mangoes: "<<endl;

cin>>mango\_no;

}

};

class apple{

protected:

int apple\_no;

public:

void getapple(){

cout<<"Enter number of apples: "<<endl;

cin>>apple\_no;

}

};

class fruit: public mango, public apple{

public:

void getfruit(){

cout<<"Total number of mangoes: "<<mango\_no<<endl;

cout<<"Total number of apples: "<<apple\_no<<endl;

cout<<"Total number of fruits: "<<mango\_no+apple\_no;

}

};

int main(){

fruit obj;

obj.getmango();

obj.getapple();

obj.getfruit();

return 0;

}

**4. Class Student contains data members RollNo and Name as protected and member**

**functions GetDetails() to get RollNo and Name and DisplayDetails() to display**

**RollNo and Name.**

**Class Marks is publicly inherited from Student. Student class contains protected data**

**member Subject1 and Subject2 i.e. marks obtained in two subjects and GetMarks()**

**and DisplayMarks() are two public member functions.**

**Class Result is publicly inherited from Marks. It contains private data member**

**TotalMarks and two public methods CalculateResult() and DisplayResult() with status**

**whether the student has “PASSED” or “FAILED”.**

**Write a C ++ program to show the results according to the following formats:**

**Enter the number of students: 2**

**Enter student roll number: 1**

**Enter name of the student: A**

**Enter the marks of subject 1: 27**

**Enter the marks of subject 2: 32**

**Enter student roll number: 2**

**Enter name of the student: B**

**Enter the marks of subject 1: 65**

**Enter the marks of subject 2: 45**

**Roll No. Name Subject1 Subject2 Total Marks Obtained Result**

**--------------------------------------------------------------------------------------------------------------------------------**

**1 A 27 32 59 FAILED**

**2 B 65 45 110 PASSED**

**----------------------------------------------------------------------------------------------------------------------------------**

**Make the necessary assumptions with comments.**

#include<iostream>

using namespace std;

class student{

protected:

char name[10];

int rollno;

public:

void getdetails(){

cout<<"Enter name: "<<endl;

cin>>name;

cout<<"Enter rollno"<<endl;

cin>>rollno;

}

void displaydetails(){

cout<<"Name and rollno is: "<<name<<" "<<rollno<<endl;

}

};

class marks: public student{

protected:

int subject1;

int subject2;

public:

void getmarks(){

cout<<"Enter sub1 marks: "<<endl;

cin>>subject1;

cout<<"Enter sub2 marks: "<<endl;

cin>>subject2;

}

void displaymarks(){

cout<<"Marks in sub1 are: "<<subject1<<endl;

cout<<"Marks in sub2 are: "<<subject2<<endl;

}

};

class result: public marks{

private:

int totalmarks;

public:

void calculateresult(){

totalmarks=subject1+subject2;

}

void displayresult(){

cout<<name<<" "<<rollno<<" "<<subject1<<" "<<subject2<<" "<<totalmarks<<endl;

if(totalmarks>=33){

cout<<"PASS"<<endl;

}

else{

cout<<"FAIL"<<endl;

}

//assumed that 33 is minimum passing marks

}

};

int main(){

int num,i;

cout<<"Enter number of students: "<<endl;

cin>>num;

result obj[num];

for(i=0;i<num;i++){

obj[i].getdetails();

obj[i].getmarks();

obj[i].displaydetails();

obj[i].displaymarks();

obj[i].calculateresult();

obj[i].displayresult();

}

return 0;

}