

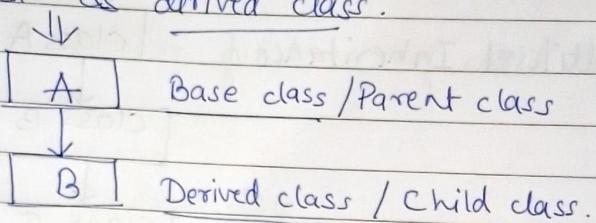
Unit 3. Extending classes using Inheritance

(16 Marks)

I) Introduction to Inheritance

* Inheritance :- "The mechanism of deriving a class from another class is known as inheritance".

- The class from which another class is derived is called "base class" while the class which is derived is called as "derived class".



- Members of class may be public, protected or private known as "access specifiers".
- The visibility of base class members within derived class is as shown in below Table 1 :

Base class member	Derived class visibility		
visibility	Public derivation	Protected derivation	Private derivation
Public members	Public	Protected	Private
Protected members	Protected	Protected	Private
Private members	Not inherited	Not inherited	Not inherited

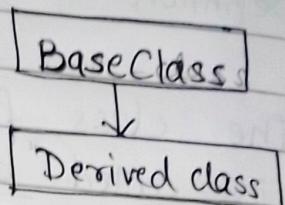
In above table,

- public derivation - The public and protected members of base class remain same visibility even in derived class where private members are never inherited.

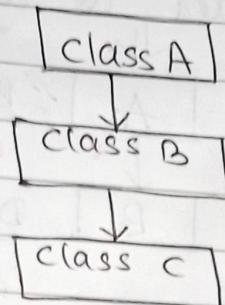
- ii) protected derivation: public & protected remain protected in derived class & private not inherited.
- iii) private derivation: All remain not inherited.

* Types of Inheritance:

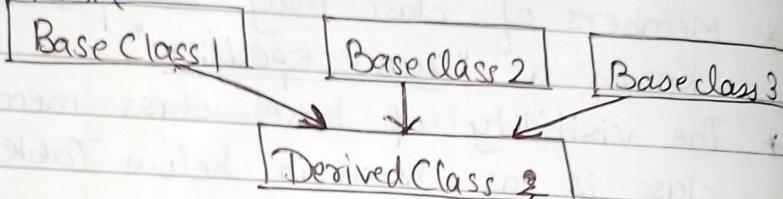
i) Single Inheritance



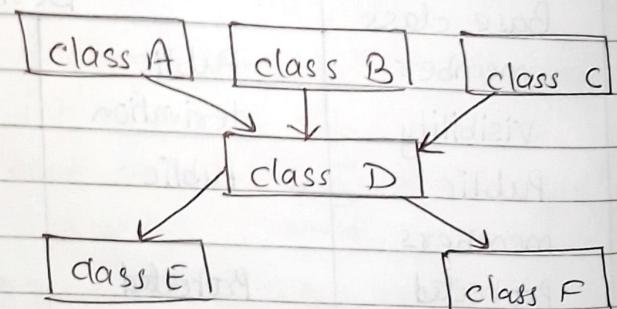
ii) Multilevel Inheritance



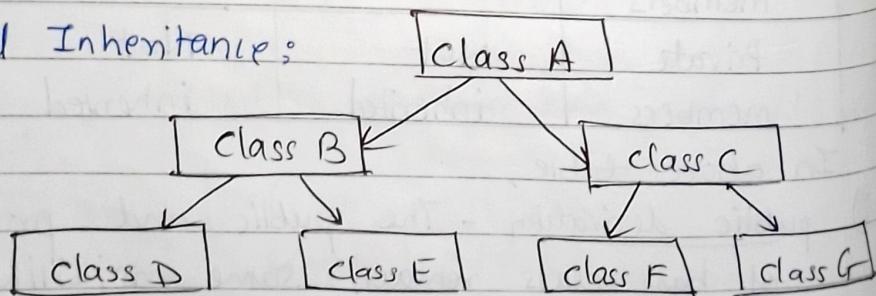
iii) Multiple Inheritance



iv) Hybrid Inheritance



v) Hierarchical Inheritance:



* Syntax: (Defining child class (derived) from parent (base) class in all types of inheritance)

class parent class
{

// members & member functions of base class
};

class child class : visibility-mode parent-class
{

// members & member functions of child class
};

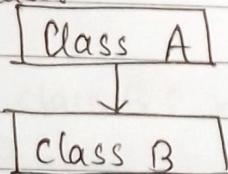
Description:

parent-class	- Base class
child-class	- Derived class
visibility-mode	- private, public, protected (access specifier)

II) * Types of Inheritance in detail

1) Single Inheritance

- In this inheritance, a single class inherits the properties of a base class.



Syntax:

class A
{

// class definition
};

class B : visibility-mode A
{

// class definition
};

(100)

• Example : #include <iostream.h>
#include <conio.h>
// Base class
class electronicDevice
{
public :
// constructor of base class
electronicDevice ()
{
cout << "I am an electronic device\n";
}
};
// derived class
class Computer : public electronicDevice
{
public :
// constructor of derived class
Computer ()
{
cout << "I am a computer\n";
}
};
// Main function
void main ()
{ // create object of derived class
Computer obj;
// constructor of base class and
// constructor of derived class will be called
getch ();
}

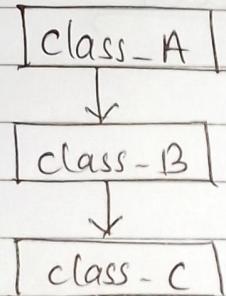
o/p:

I am an electronic device
I am a computer.

- Explanation:
- The subclass Computer inherits the base class electronicDevice in a public mode.
 - So all the public and protected member function and data members of class electronicDevice are directly accessible to class Computer.
 - As single derived class inherits single base class Hence, its Single Inheritance.

2) Multilevel Inheritance

- The inheritance in which a class can be derived from another derived class is known as Multilevel inheritance.



- Syntax:

```
class class-A
```

```
{ // class def"
```

```
}
```

```
class class-B : visibility-mode class-A
```

```
{ // class def"
```

```
}
```

```
class class-C : visibility-mode class-B
```

```
{ // class def"
```

```
}
```

• Example :

WAP to calculate percentage of a student using multi level inheritance. The base class function will accept the marks in three subjects from user. A class will be derived from the above mentioned class that will have a function to find the total marks obtained and another class derived from this will have functions to calculate and display the percentage scored. (4 Marks)

```
# include <iostream.h>
```

```
# include <conio.h>
```

```
class Data
```

// Base class

```
{
```

protected :

```
int p,c,m;
```

public :

```
void read()
```

```
{
```

```
cout << "Enter marks obtained in Physics , chemistry  
and Maths";
```

```
cin >> p >> c >> m;
```

```
}
```

```
};
```

```
class Sum : public Data
```

// Derived class

```
{
```

protected :

```
int total ;
```

public :

```
void sum()
```

```
{
```

$$\text{total} = p + c + m;$$

```
}
```

```
};
```

(103)

class Percent : public Sum // Derived class 2

{

private :

float percent;

public :

void calculate()

{

$$\text{percentage} = \text{total} / 300.0 * 100;$$

void display()

{

$$\text{cout} << \text{"The percentage is"} << \text{percentage};$$

}

};

void main()

{

clrscr();

Percent a;

a.read();

a.sum();

a.calculate();

a.display();

getch();

}

// Main fun

o/p: Enter marks obtained in Physics, Chemistry and Maths 90

98

99

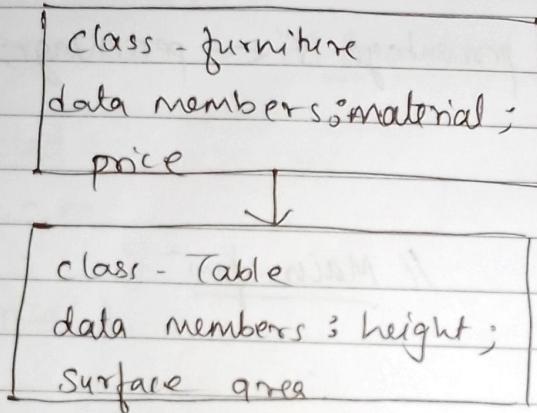
The percentage is : 95.666664

3. Multiple Inheritance)

1) Single Inheritance:

WAP to add 2 numbers using single inheritance such that the base class function must accept the 2 numbers from the user and the derived class function must add these numbers and display the sum.

2) WAP to implement single inheritance from following fig:-



3) WAP to demonstrate constructor in derived class with respect to order of calling constructor and passing parameter to base class constructor.

4) Multilevel Inheritance :- WAP to identify the type of inheritance shown in following figure.

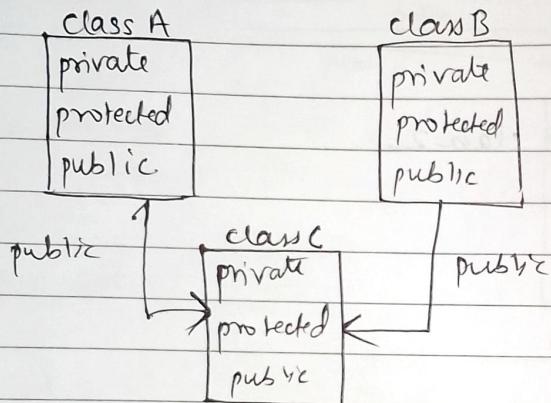
Implement it by using suitable member function.



Q) Write a C++ program to declare a class College with members as college code. Derive a new class as STUDENT with members as student id. Accept and display details of student along with college for one object of student.

3) Multiple Inheritance

- In this case one class is derived from multiple classes.



Syntax:

```
class Base-class1
```

```
{  
  // class def  
}
```

```
class Base-class2
```

```
{  
  // class def  
}
```

Class Derived class : visibility_mode_1 Base-class1 , visibility_mode_2 Base-class2

```
{  
  // class def  
};
```

• Example :

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
// class A : Base class-1
```

```
class electronicDevice
```

```
{
```

```
public:
```

```
electronicDevice()
```

```
{
```

```
cout << "I am an electronic device\n";
```

```
}
```

```
};
```

```
// class B : Base class-2
```

```
class Computer
```

```
{
```

```
public:
```

```
Computer()
```

```
{
```

```
cout << "I am a computer";
```

```
}
```

```
};
```

```
// class C : Derived class
```

```
class Linux-based : public electronicDevice, public  
Computer
```

```
{ };
```

```
void main()
```

```
{ clrscr();
```

```
Linux-based obj;
```

```
getch();
```

```
}
```

```
O/p
```

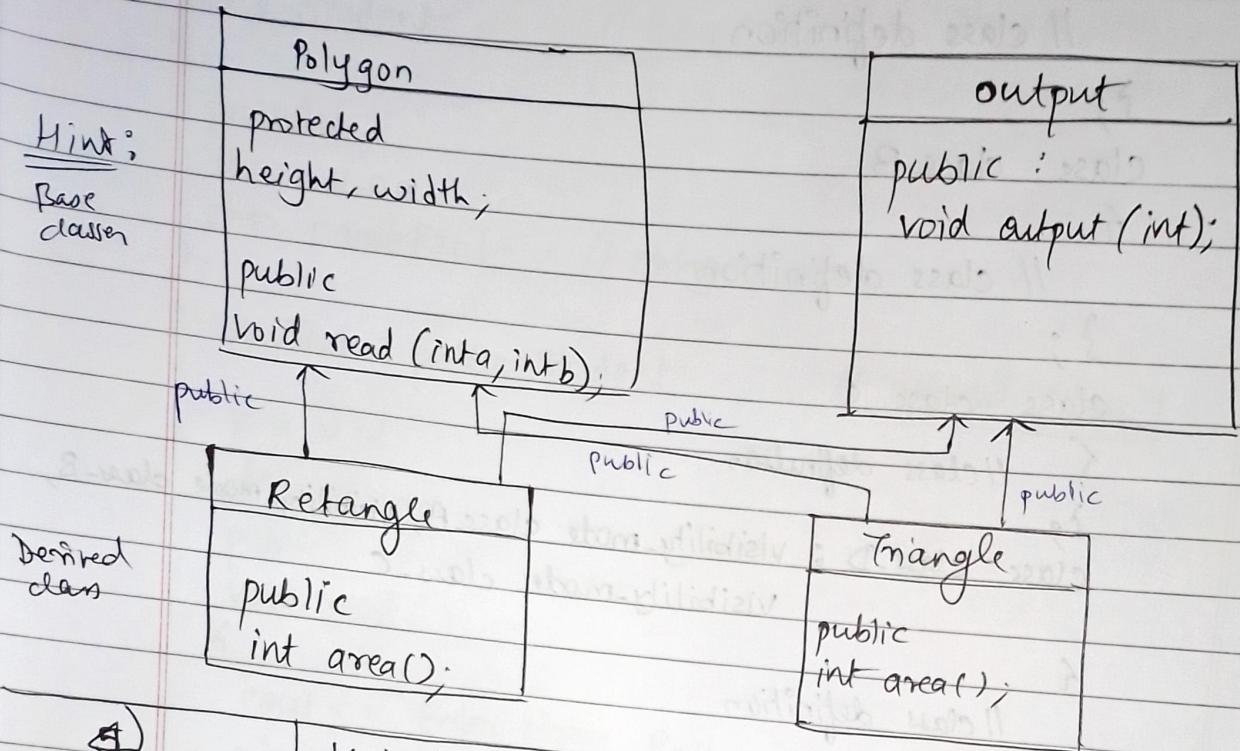
I am an electronic device

I am a computer.

107

Try It

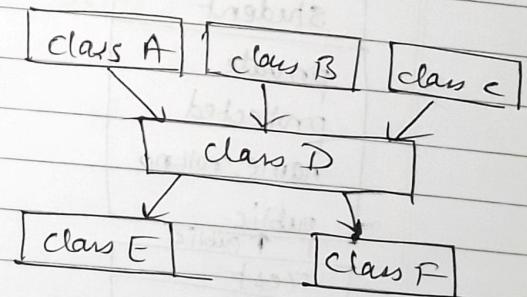
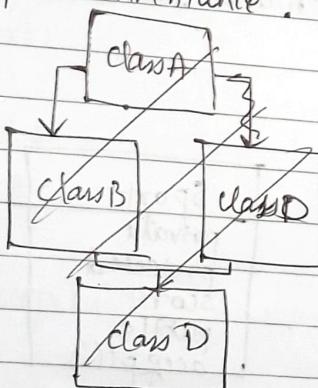
- 1) WAP to define the following relationship using multi inheritance:-



4)

Hybrid Inheritance

- In this case there is mixture of multi level and multiple inheritance. Hence named Hybrid.



- In this case there is a mixture of multi level and multiple inheritance. hence class **E** is derived from class **D** & class **D** is derived from class **A**. Hence multilevel: class **D** is derived from class **A**, class **B** & class **C** hence multiple.

• Syntax:

```
class class-A
```

{

```
// class definition
```

};

```
class class-B
```

{

```
// class definition
```

};

```
class class-C
```

{

```
// class definition
```

};

```
class class-D : visibility-made class-A, visibility-made class-B,  
visibility-made class-C
```

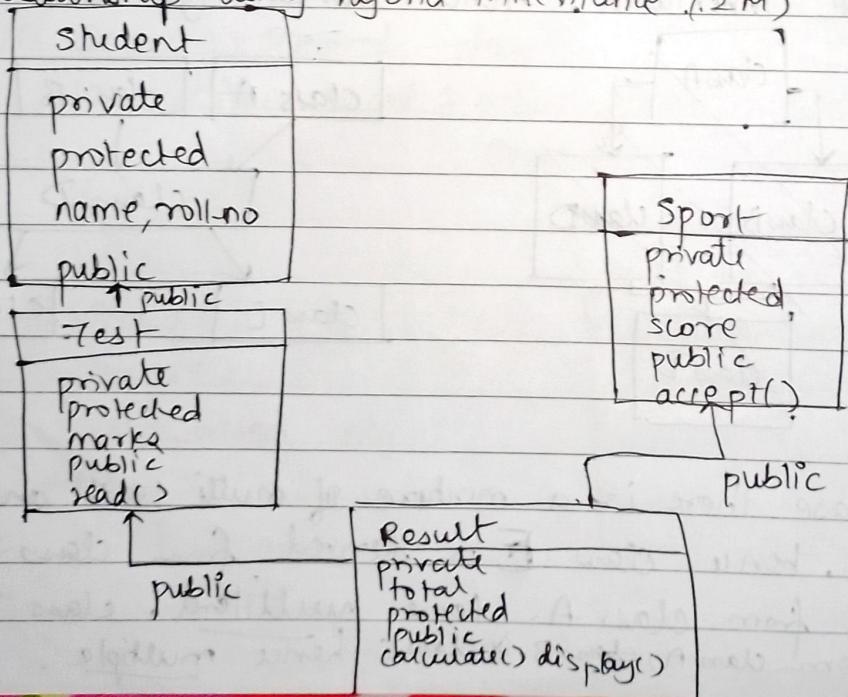
{

```
// class definition
```

};

• Example:

- Q) Write a program to define the following relationship using hybrid inheritance (2 M)



```
#include <iostream.h>
#include <conio.h>
#include <stdio.h> // #include <string.h>
class Student // BaseClass 1
{
```

```
protected :
int roll-no;
char name[20]; // string name
};
```

```
class Test : public Student // Derived class 1
{
```

```
protected :
```

```
int marks
```

```
public :
```

```
void read()
```

```
{
```

```
cout << "Enter Name, Roll-no, Marks of Student : \n";
cin >> name >> roll-no >> marks;
```

```
gets(name) // cin >> name // getline (cin, fullname);
```

```
}
```

```
class Sports
```

```
{
```

```
protected :
```

```
int score;
```

```
public :
```

```
void accept()
```

```
{
```

```
Cout << " 1. Student has won in national sports event \n
2. Student has not won in any national sports \n
Enter your choice ";
```

```
cin >> score;
```

```
{
```

class Result : public Test, public Sports

{
int total;

public:

void calculate()

{

if(score == 1)

total = marks + 15;

else

total = marks;

}

void display()

{

cout << "The total is" << total;

}

};

void main()

{

clrscr();

Result r;

r.read();

r.accept();

r.calculate();

r.display();

r.getch();

}

Enter name, roll-no and marks obtained :

Sandeep

24

81

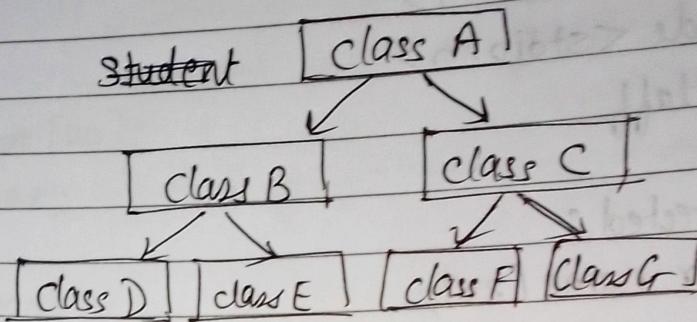
1. Student has won in national sports event.

2. Student has not won in -1-

Enter your choice : 1

The total is 96

5. Hierarchical Inheritance:



- When multiple classes are derived from a class and further more classes are derived from these derived classes.

- Syntax : class A

```
{ // class def  
};
```

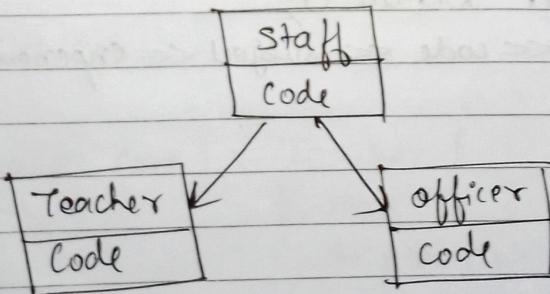
```
class B : visibility-mode A  
{
```

```
// class def  
};
```

```
class C : visibility-mode A  
{ // class def  
};
```

- Example :

- WAP to implement inheritance as shown in Figure.
Assume suitable member function.



```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
```

```
class Staff
```

```
{
```

```
protected:
```

```
char name[20];
```

```
int code;
```

```
};
```

```
class Teacher : public Staff
```

```
{
```

```
private:
```

```
char subject[20];
```

```
int exp;
```

```
public:
```

```
void read()
```

```
{
```

```
cout << " Enter name, code, subject and exp  
of teacher :";
```

```
cin >> code >> exp;
```

```
gets(name);
```

```
gets(subject);
```

```
}
```

```
void display()
```

```
{ cout << " Teacher Details :";
```

```
cout << name << code << subject << exp;
```

```
?
```

```
};
```

class Officer : public Staff

{

private :

char dept[20];

int grade;

public :

void read()

{

cout << " Enter name, code, dept and grade of officer ";

gets(name);

cin >> code;

gets(dept);

cin >> grade;

}

void display()

{ cout << " Officer Details : " ;

cout << name << code << dept << grade ;

}

};

void main()

{ clrscr();

int choice;

cout << " 1. Teacher \n 2. Officer \n Enter the choice,
whose details you want to enter : " ;

cin >> choice

switch(choice)

{

case 1 : Teacher t ;

t.read();

t.display();

break ;

(114)

case 2: Officer o;
o. read();
o. display();
break;

default : cout << "Invalid choice";

getch();

o/p :
1. Teacher
2. Officer

Enter the choice, whose details you want
to enter: 1

Enter name, code, subject and exp of teacher:

Satish

234

Maths

13

Teacher Details:-

Satish 234 Maths 13

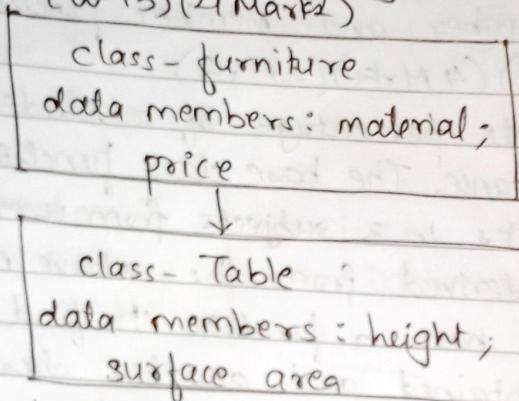
IMP Questions

- Q1) What is inheritance? Why inheritance is used in C++? (W-16, W-19, 2 Marks)
- Q2) State and explain various visibility modifiers in inheritance (W-16, W-19, 4 Marks).
- Q3) Write syntax to define derived class (W-18)
(2 M)
- Q4) Describe derived class with example. (3-19, 2 Marks)
- Q5) Explain various types of inheritance with example (W-16, 8 Marks)

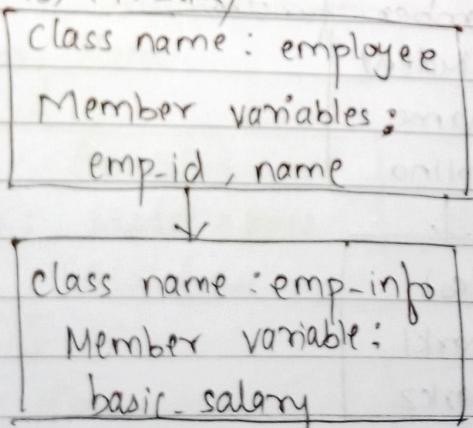
Single Inheritance:

- Q9) WAP to add 2 no using single inheritance such that the base class function must accept the 2 no from the user and the derived class function must add these no's and display the sum.
(W-16, S-17)(4 Marks)

- Q10) WAP to implement single inheritance from following figure. accept and display the data for one table.
(W-15)(4 Marks)



- Q11) WAP to implement single inheritance. Declare base class 'Employee' with emp-no and emp-name. Declare derived class 'Fitness' with height and weight. Accept and display data for one employee. (W-16, 4 Marks)
- Q12) WAP to implement single inheritance from following figure:- (W-19) (4 Marks)



- Q13) Describe with examples, passing parameters to base class constructor and derived class constructor by creating object derived class. (S-19)(4 Marks)

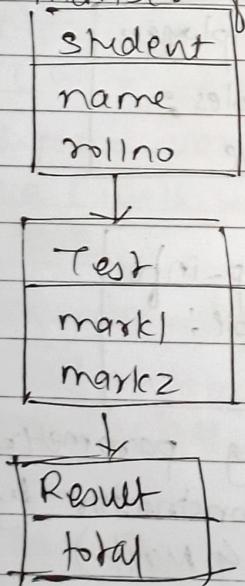
Q14) WAP to demonstrate constructor in derived class with respect to order of calling constructor and passing parameters to base class constructor.
(S-15, W-15, S-16, W-17)(4 Marks)

Multilevel:

Q15) What is multilevel inheritance? Draw the diagram to show multilevel inheritance using classes with data member and member function.
(S-15, W-19)(4 Marks)(2 Marks)

Q16) WAP to calculate percentage of a student using multilevel inheritance. The base class function will accept the marks in 3 subjects from user. A class will be derived from the above mentioned class that will have a function to find the total marks obtained and another class derived from this will have functions to calculate and display the percentage scored.
(S-16)(4 Marks)

Q17) Identify the type of inheritance shown in following figure. Implement it by using suitable member function.



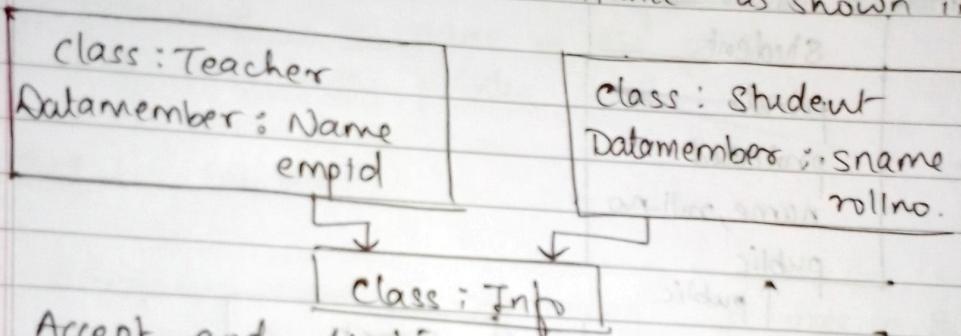
Q17) WAP to declare a class "College" with members as college code. Derive a new class as Student with members as stupid. Accept and display details of student along with college for one object of student. (5-19)(2 Marks)

Multiple Inheritance + Multilevel Inheritance :

Q18) Explain multiple inheritance. (W-17) (4 Marks)

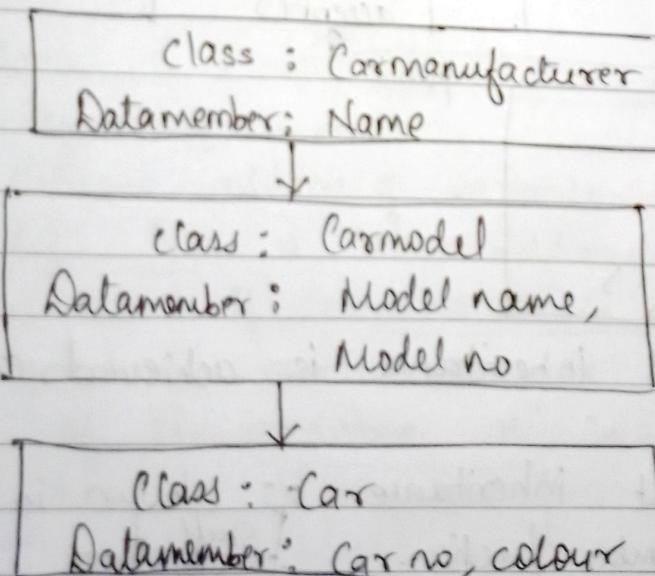
Q19) WAP to define the following relationship using multiple inheritance.

Q20) WAP to implement inheritance as shown in figure:-



Accept and display data of one teacher and one student using object of class 'Info'. (W-18, 4 Marks)

Q21) WAP for following multilevel inheritance



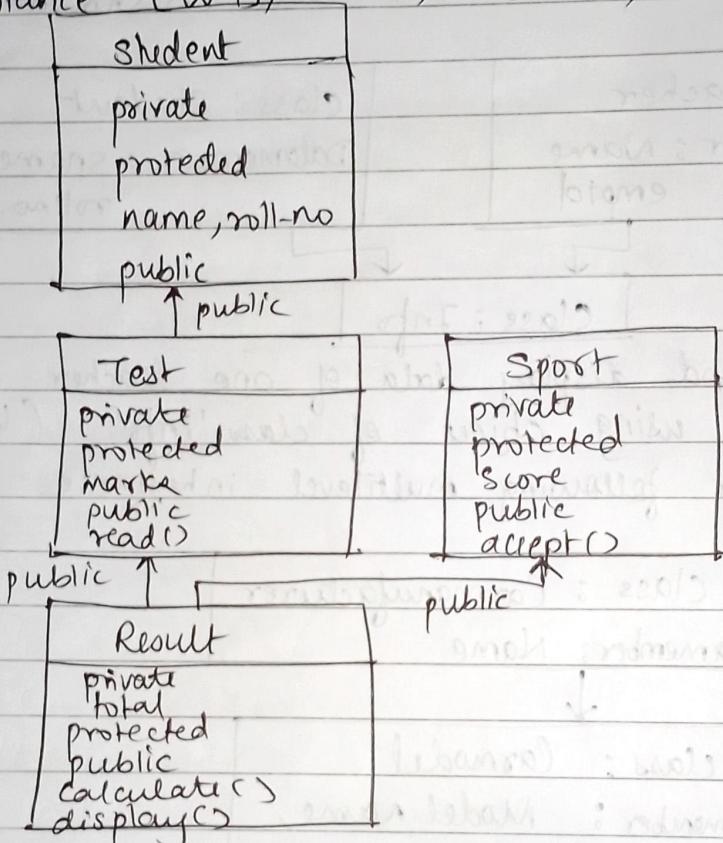
Accept and display data for one car with all details.

- Q22) WAP to declare a class student with members as roll no., name and department. Declare a parameterized constructor with default value for department as 'co' to initialize members of object. Initialize and display data for two students.
(S-19)(6 Marks)

Hybrid Inheritance :

- Q23) What is hybrid inheritance? Give one example.
(S-16)(4 Marks)

- Q24) WAP to define the following relationship using hybrid inheritance (W-19, 2 Marks) or (8 Marks)



- Q25) How hierarchical inheritance is achieved, explain with example.

- Q26) WAP to implement inheritance as shown in figure
Assume suitable member function.
(S-17, 8 Marks)

