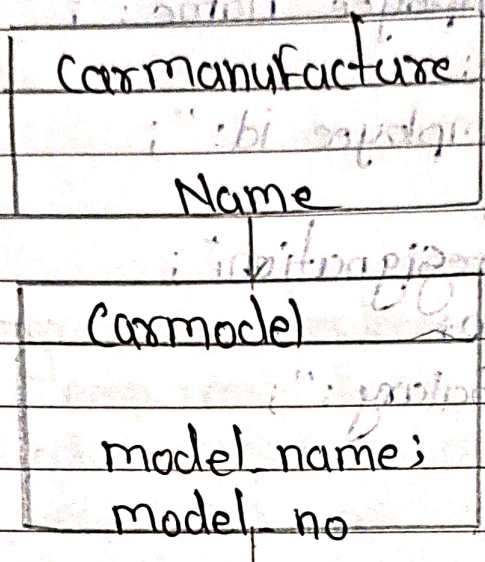


Q6 write a C++ program for following multilevel inheritance.



→

```

#include <iostream.h>
#include <conio.h>
class CarManufacture
{
public:
char Name [50];
};

class CarModel : public CarManufacture
{
public:
char model_name;
int model_no;
};
  
```

```
class car : public CarManufacture, Public CarModel
```

```
{
public:
int car_no;
char colour[50];
```

```
void
class car : public CarModel
```

```
{
public:
int car_no;
char colour[50];
```

```
void accept()
```

```
{
cout << "Name:";
cin >> name;
cout << "Enter car model:";
cin >> model_name;
cout << "Enter model no:";
cin >> model_no;
cout << "Enter car no & colour";
cin >> car_no >> colour;
}
```

```
void display()
```

```
{
cout << "Name:" << name;
cout << "model Name:" << model_name;
cout << "model No:" << model_no;
cout << "Car No:" << car_no;
}
```

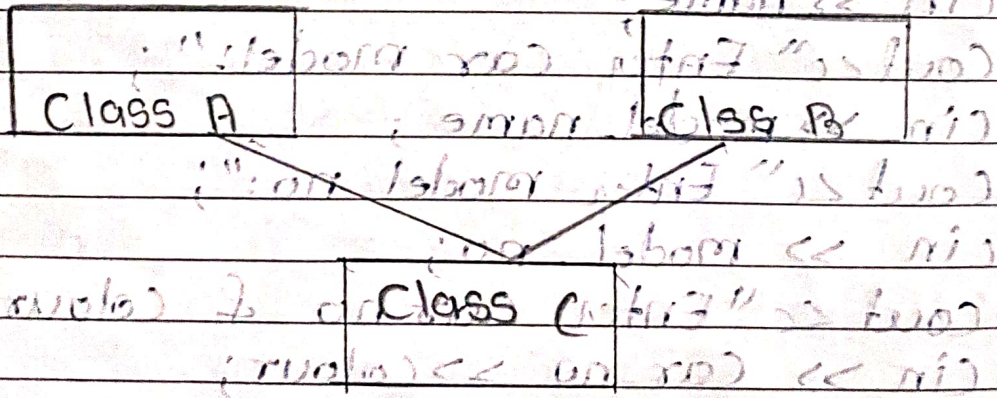
```

cout << "Car colour:" << colour;
}
};

void main()
{
    Car c;
    c.accept();
    c.display();
}
    
```

3] Multiple inheritance

In this case one class is derived from multiple classes



To derive a class from multiple classes we need to use the following syntax

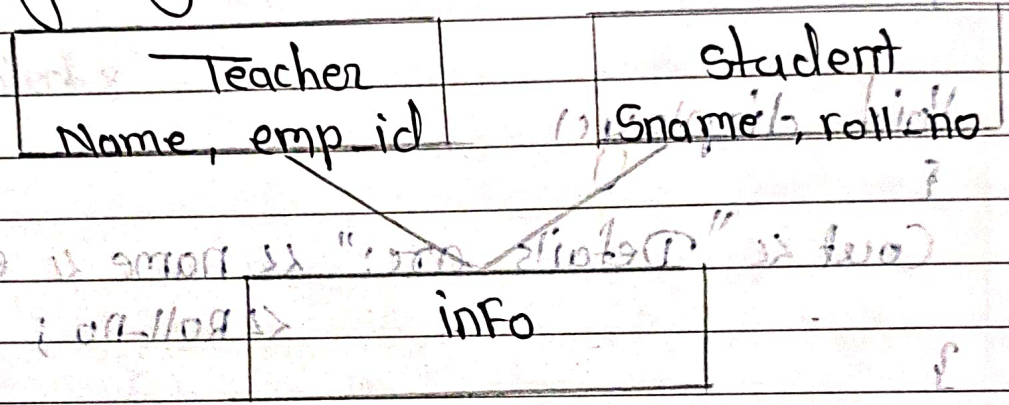
```

class derived : access specifier parent class name 1,
class name
access specifier 2 parent class name 2 ;
    
```

In the above diagram class c is derived from class B as well as class A

Class c: Public A, Public B

Q1. Write a program to implement inheritance shown in following figure.



```

#include <iostream.h>
#include <conio.h>
class Teacher
{
public:
    char name[50];
    int emp_id;
};
    
```

```

class Student
{
public:
    char Sname[50];
    int roll no;
};
    
```

Info

Class info : public teacher, public student

```

public:
    void accept()
    {
        cout << "Enter details" << endl;
        cin >> name >> emp_id >> Sname >> Roll-no;
    }

    void display()
    {
        cout << "Details are:" << name << emp_id << Sname
            << Roll-no;
    }
};

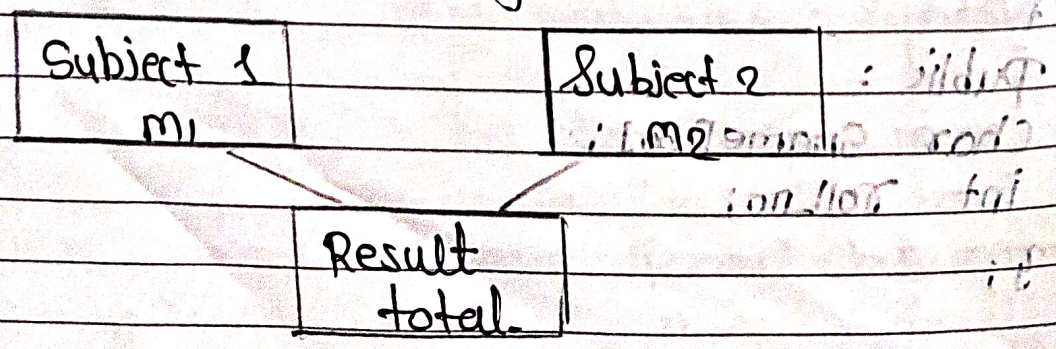
```

```

void main()
{
    info i;
    i.accept();
    i.display();
}

```

Q2. Develop a program to implement inheritance shown in following figure.





```

#include <iostream.h>
#include <conio.h>
class Subject 1
{
public:
int m1;
};

class Subject 2
{
public:
int m2;
};

class Result : public Subject 1, public Subject 2
{
public:
int total;
void accept()
{
cout << "Enter marks m1 & m2:";
cin >> m1 >> m2;
}

void compute()
{
total = m1 + m2;
}

void display()

```

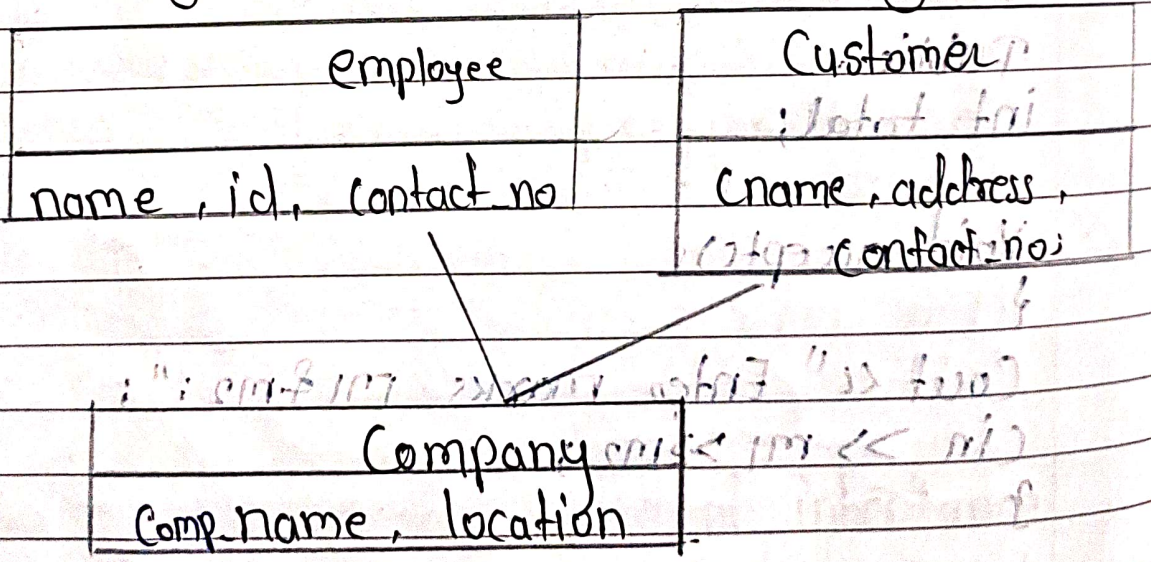
```

}
cout << "total marks : " << total;
}
};

void main()
{
total
Result r;
r.accept();
r.compute();
r.display();
}

```

Q3. Write a program to implement: Following inheritance.



```

-> #include <iostream.h>
#include <conio.h>
class employee
{
public:
char name[20];

```

```
int id, Contact_no;
};
```

```
class customer
```

```
{
public:
char cname [50], address [50];
int contact_no;
};
```

```
class Company : public employee, public customer
```

```
{
public:
char name [50], location [50];
void accept ()
{
cout << "Enter company details:";
cin >> name >> id >> Contact_no >> "Customer name" >> address;
>> cont_no >> comp_name >> location;
}
```

```
void display ()
{
cout << "Company Details are:";
cout << "Employee Name" << name;
cout << "Employee id" << id;
cout << "Contact no" << contact_no;
cout << "Customer name" << cname;
cout << "Customer Address" << address;
cout << "Customer Contact_no" << cont_no;
}
```



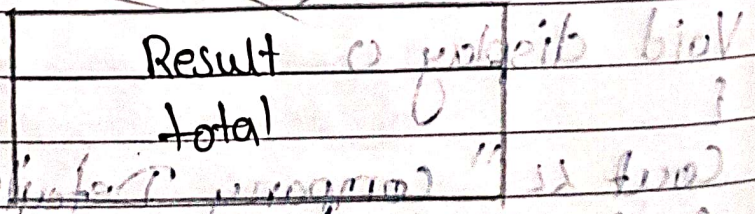
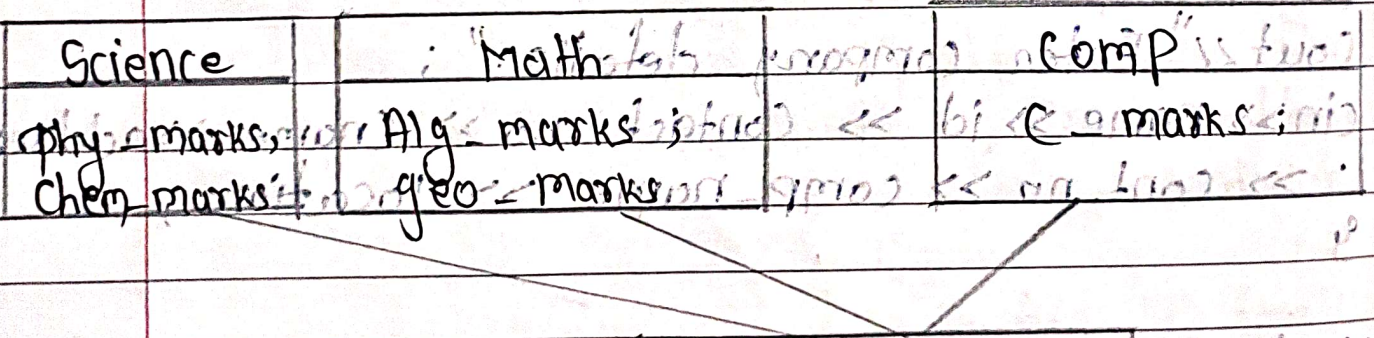
```

cout << "Company name:" << comp.name;
cout << "location:" << location;
}
};

void main()
{
    Company c;
    c.accept();
    c.display();
}

```

Q.4 Write a program to implement following inheritance



```

-> #include <iostream.h>
#include <conio.h>
class Science
{
public:

```

```
int phy_marks, chem_marks;  
};
```

```
class math  
{  
public:  
int Alg_marks, geo_marks;  
};
```

```
class comp  
{  
public:  
int C_marks;  
};
```

```
class Result: public Science, public math, public comp
```

```
{  
void public:  
int total;
```

```
void accept()
```

```
{  
cout << "Enter marks: ";  
cin >> Alg_marks >> geo_marks >> phy_marks >>  
chem_marks >> C_marks;  
};
```

```
void calculate()
```

```
{  
total = phy_marks + chem_marks + Alg_marks + geo_marks +  
C_marks;  
};
```

```

void display() {
    cout << "Total marks:" << total;
}
}

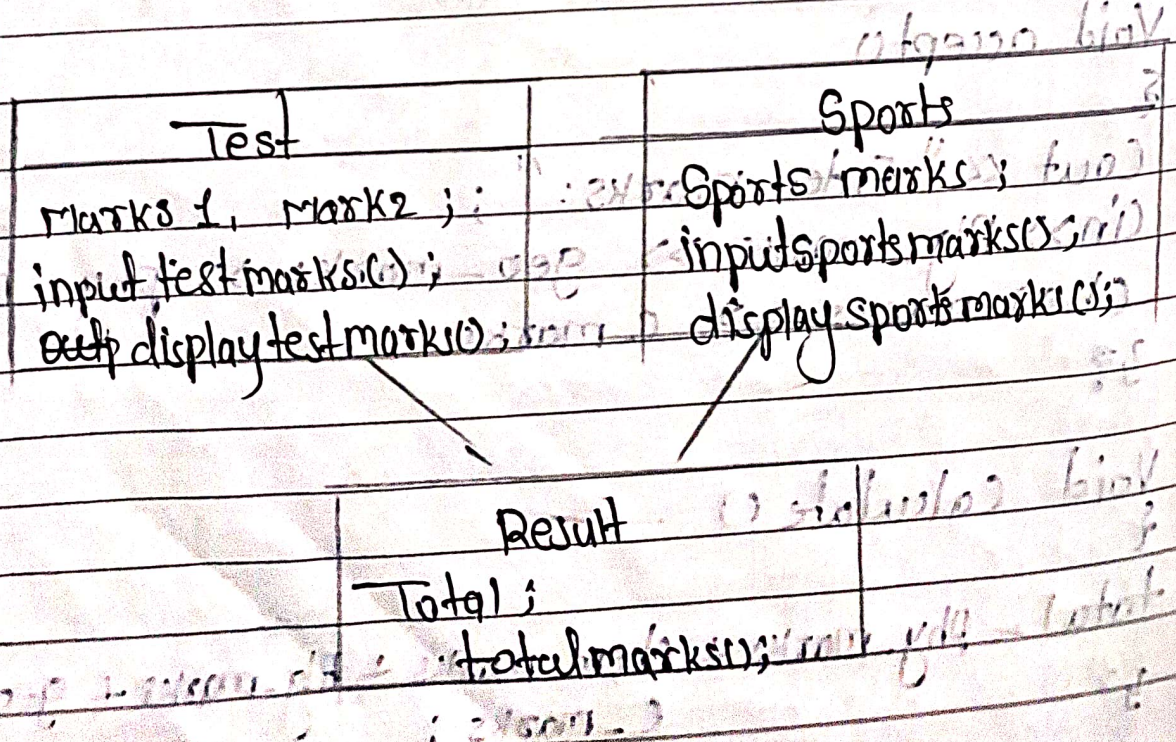
```

```

void main() {
    Result r;
    r.accept();
    r.calculate();
    r.display();
}

```

Q.5 Write a program to implement multiple inheritance as shown in figure accept & display data of Test marks & Sport marks using object of class Result



```
#include <iostream.h>
#include <conio.h>
class test
{
public:
int mark1, mark2;
inputtestmarks()
{
cout << "Enter marks:";
cin >> mark1 >> mark2;
}

displaytestmarks()
{
cout << "Test marks: " << mark1 << mark2;
}
};
```

```
class sports
{
public:
int sports marks;
inputsportsmarks()
{
cout << "Enter sports marks:";
cin >> sports marks;
}
};
```

```
display Sportsmarks()
{
    cout << "Sports marks is : " << Sport_marks;
}
};
```

```
class Result : Public Test, Public Sports
{
public:
    int total;
```

```
total marks()
{
    total = Marks 1 + marks 2 + Sports_marks;
    cout << "Total marks is : " << total;
}
};
```

```
Void main()
{
    Result r;
    r.inputtestmarks();
    r.inputSportsmarks();
    r.displaytestmark();
    r.displaySportmark();
    r.totalmarks();
    getch();
}
```

Q6. Difference between multiple & multilevel inheritance

Multiple inheritance

Multilevel inheritance

i] It is an inheritance type where a class inherits from more than one base class.

i] It is an inheritance type that inherits from derived class making that derived class a base class for a new class.

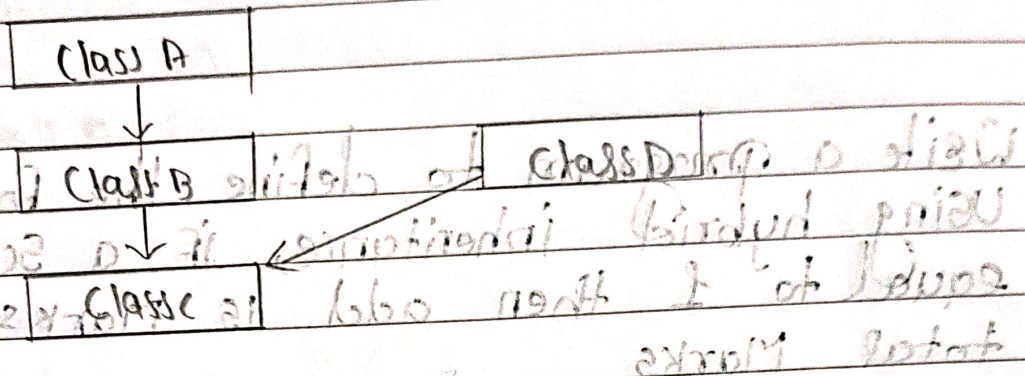
ii] Multiple inheritance is not widely used because it makes the system more complex.

ii] Multilevel inheritance is widely used.

iii] Multiple inheritance has two class levels namely base class & derived class

iii] Multilevel inheritance has 3 class level namely base class, intermediate class, and derived class

* Hybrid inheritance
 In this case there is a mixture of multilevel and multiple inheritance.



Syntax:

```

Class A
{
}

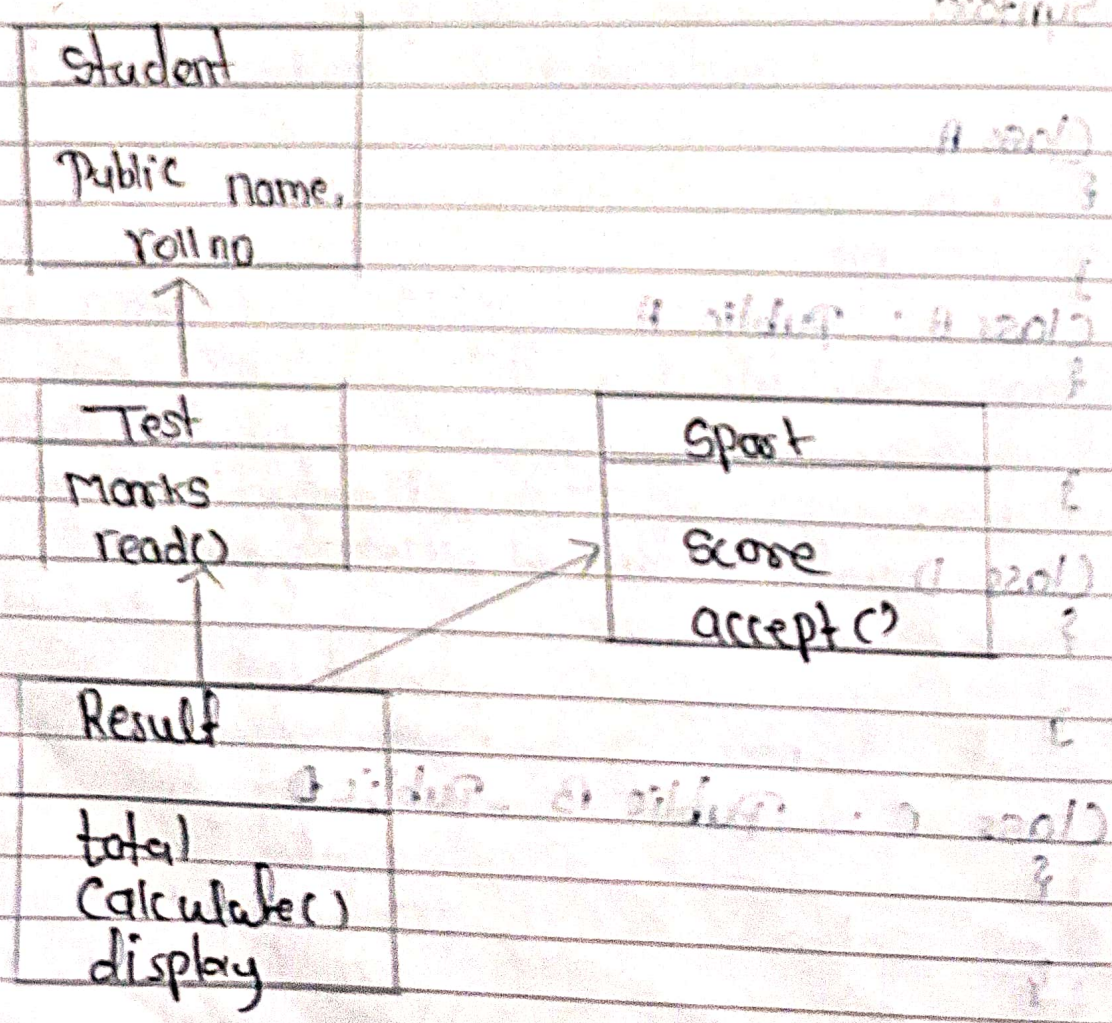
Class B : Public A
{
}

Class D
{
}

Class C : Public B , Public D
{
}
    
```

In this case class C is derived from class B which is derived from class A hence it is a multilevel inheritance, class C is also derived from class D. Hence it is a multiple inheritance.

Q. Write a program to define the following relationship using hybrid inheritance, if a score is exactly equal to 1 then add 15 marks to the total marks




```

#include <iostream.h>
#include <conio.h>
class student
{
public:
char name[50];
int Rollno;
};

```

```

class test
{
public:
int marks;
void read()
{
cout << "Enter marks, rollno and name!";
cin >> marks >> name >> rollno;
};
};

```

```

class sports
{
public:
int score;
void accept()
{
cout << "1. student has won the national sports"
2. student has not won the national sports";
cout << "enter your choice between 1 & 2";
cin >> score;
};
};

```

Class Result : Public test, Public sport

```

{
Public :
int total;
Void calculate ()
{
    if (score == 1)
    {
        total = marks + 15;
    }
    else
    {
        total = marks;
    }
}
Void display ()
{
    cout << "total marks" << total;
}
};

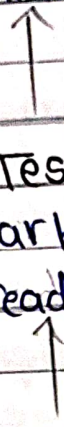
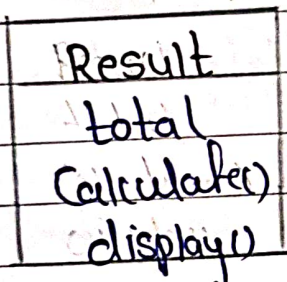
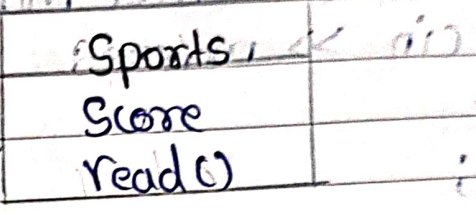
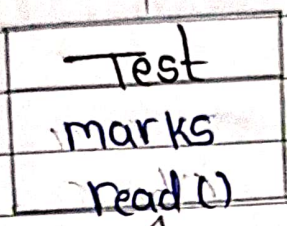
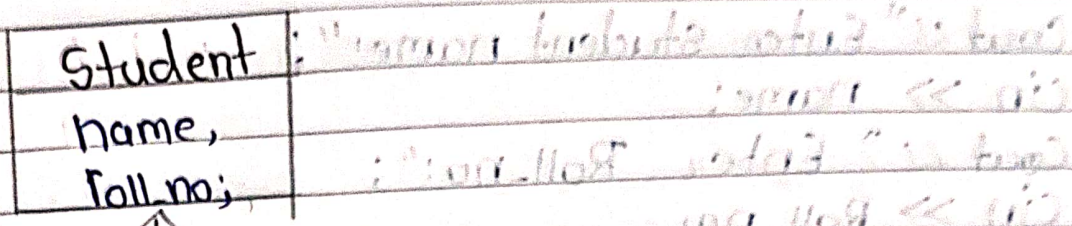
```

```

Void main ()
{
    Result r;
    r.read ();
    r.accept ();
    r.calculate ();
    r.display ();
    getch ();
}

```

H.W



```
#include <iostream.h>
#include <conio.h>
class Student
```

```
{
public:
char name[50];
int roll no;
```

```
class test : public Student
```

```
{
public:
int marks;
void read()
}
```

```
class Sports
{
public:
int Score;
void read()
}
```

```

cout << "Enter student name:";
cin >> name;
cout << "Enter Roll no:";
cin >> Roll no;
cout << "Enter marks:";
cin >> marks;
}
};

```

```

Class Sports
{
public:
int score;
void read()
{

```

cout << "1. Student has own the national sports in 2. Student has not own the national sports"

```

cout << "Enter choice 1 & 2";
cin >> score;
}
};

```

```

Class result : public test , public Sports
{
public:
int total;
void calculate()
{
if (score == 1)
{

```

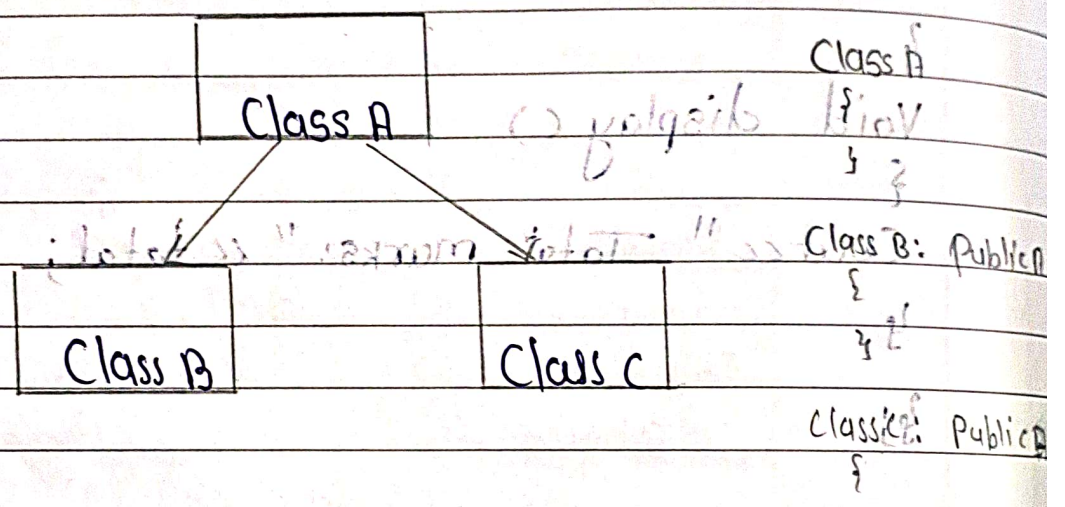
```
total = marks + 15;
}
else
{
total = marks;
}
```

```
void display()
{
cout << " Total marks: " << total;
}
};
```

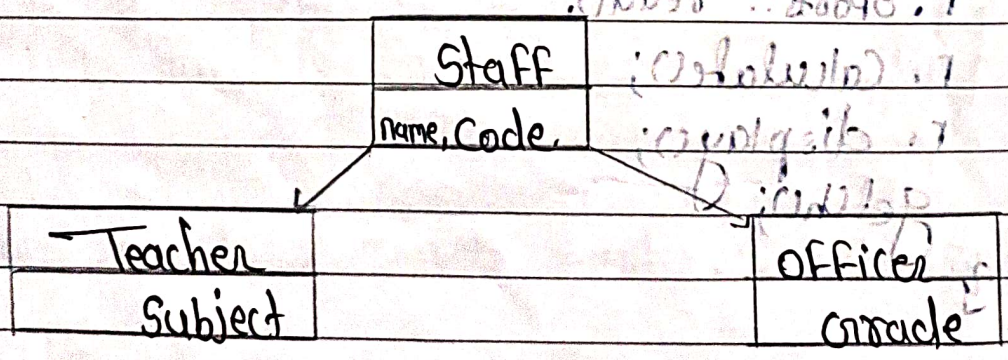
```
void main()
{
result r;
r.test :: read();
r.sports :: read();
r.calculate();
r.display();
getch();
}
```

5] Hierarchical inheritance :: 314 210011 - 10101

When multiple classes are derived from a class & further more classes are derived from these derived classes, is called as Hierarchical inheritance.



Q. Write a program to implement inheritance as shown in fig. Assume suitable member function.



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

```
class Staff
{
public:
int code;
char name[50];
};
```

```
class teacher : public Staff
{
```

```
public:
char Subject [20];
void read()
{
cout << "Enter Code , name & Sub:";
cin >> code >> name >> subject;
}
```

```
void display()
{
cout << "teacher's data are:" << code << name <<
subject;
}
```

```
class : Officer : public Staff
{
public:
int grade;
void accept()
{
```

```
cout << "enter grade";  
cin >> grade;  
}
```

```
void show()  
{  
    cout << "grade is:" << grade;  
}
```

```
void main()  
{  
    teacher t;  
    t.read();  
    t.display();  
    officer o;  
    o.accept();  
    o.show();  
}
```

[using switch case]

```
void main()  
{  
    int choice;  
    cout << "1. teacher 2. officer";  
    cout << "Enter your choice:";  
    cin >> choice;  
    switch (choice)  
    {
```



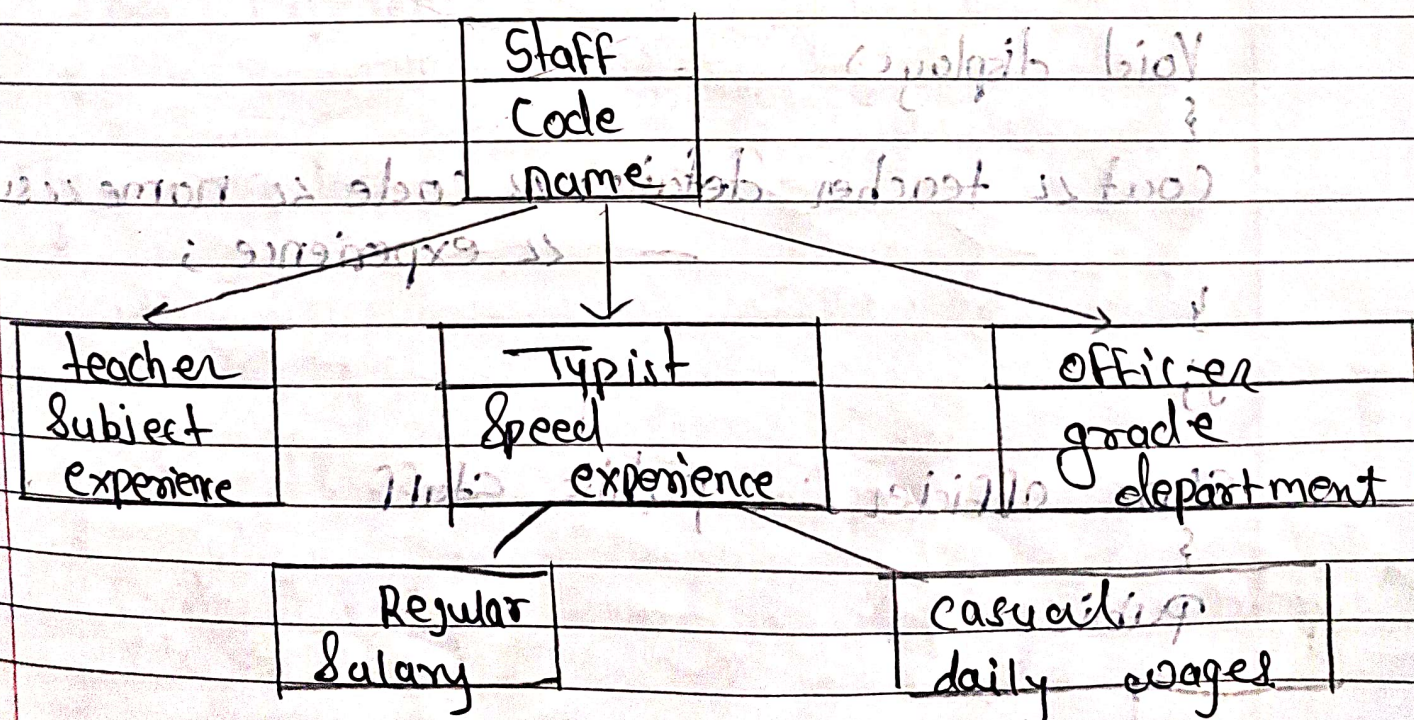
```

Case 1:
teacher t;
t.read();
t.display();
break;
    
```

```

Case 2:
officer o;
o.accept();
o.show();
default:
cout << "invalid choice"
}
    
```

Q2: Write a program to define the following inheritance relationship:



```
#include <iostream.h>
#include <conio.h>
class staff
{
public:
int code;
char name[50];
};
```

```
class teacher : public staff
{
char subject[50];
int experience;
```

```
void accept()
{
cout << "Enter code, name, subject & experience:";
cin >> code >> name >> subject >> experience;
}
```

```
void display()
{
cout << "teacher details:" << code << name << subject
<< experience;
}
```

```
class officer : public staff
{
public:
```



```
int grade;
char department [50];
```

```
void read()
{
```

```
cout << "Enter grade & department :";
```

```
cin >> grade >> department;
```

```
void show()
{
```

```
cout << "office details : " << grade << department ;
```

```
}
```

```
};
```

```
class typist : public staff {
```

```
{
```

```
public:
```

```
int speed;
```

```
char experience [50];
```

```
};
```

```
class regular : public typist
```

```
{
```

```
public:
```

```
int salary;
```

```
void get()
{
```

```
cout << "Enter speed & experience & salary :";
```

```
cin >> speed >> experience >> salary;
```

```
void set()
{
    cout << "Regular typiest" << speed << experience
    << salary;
}
```

```
class Casual : public typiest
```

```
{
public:
    int daily wages;
```

```
void getdata()
{
```

```
    cout << "Enter daily wages";
```

```
    cin >> daily wages;
}
```

```
void setdata()
{
```

```
    cout << "Daily wages:" << daily wages;
}
```

```
};
```

```
void main()
{
```

```
    int choice;
```

```
    cout << "1. teacher In 2. officer In 3. Regularly
    4. Casual In";
```

```
    cout << "Enter your choice:";
```

```
    cin >> choice;
```

Switch (choice)

```
{
  case 1:
    teacher t;
    t.accept();
    t.display();
    break;
```

```
Case 2:
  officier o;
  o.read();
  o.show();
  break;
```

```
Case 3:
  Regular r;
  r.get();
  r.set();
  break;
```

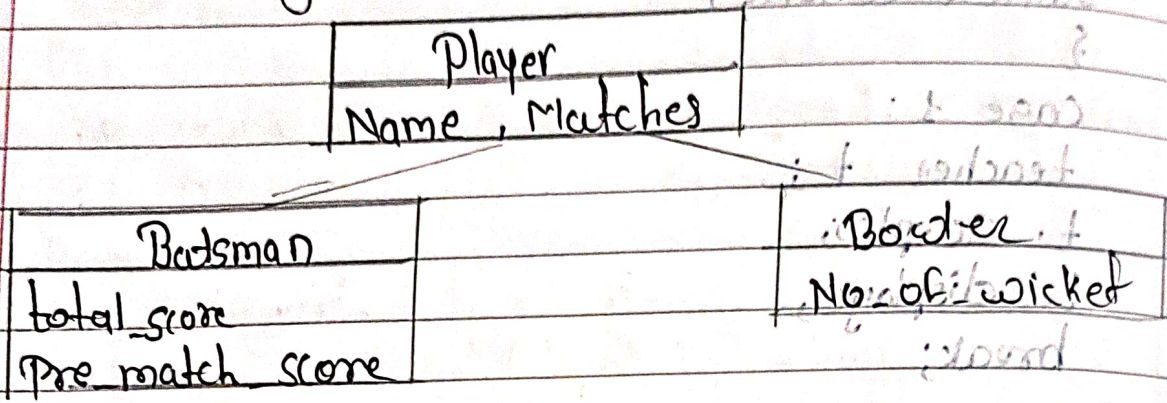
```
Case 4:
  Casual c;
  c.getdata();
  c.setdata();
  break;
```

default:

cout << "Invalid choice" << endl;

```
}
// ...
}
```

Q.1 Write a program to define following inheritance



```

#include <iostream.h>
#include <conio.h>
class Player
{
public:
    char Name[50];
    int matches;
};

class Batsman : public Player
{
public:
    int total score;
    int pre match score;
};

void read()
{
    cout << "Enter Name, matches, total score &
    pre-match score:";
    cin >> Name >> matches >> total score >>
    pre match score;
}
  
```

```
void show()
{
    cout << "Batsman Details:" << name << matches
    << total score << pre-match score;
}
```

Write a program to show following details of batsman

```
class Bowler : public Player
{
public:
    int No. of wickets;
```

```
void accept()
{
    cout << "Enter No. of wickets:";
    cin >> No. of wickets;
}
```

```
void display()
{
    cout << "wickets:" << No. of wickets;
}
};
```

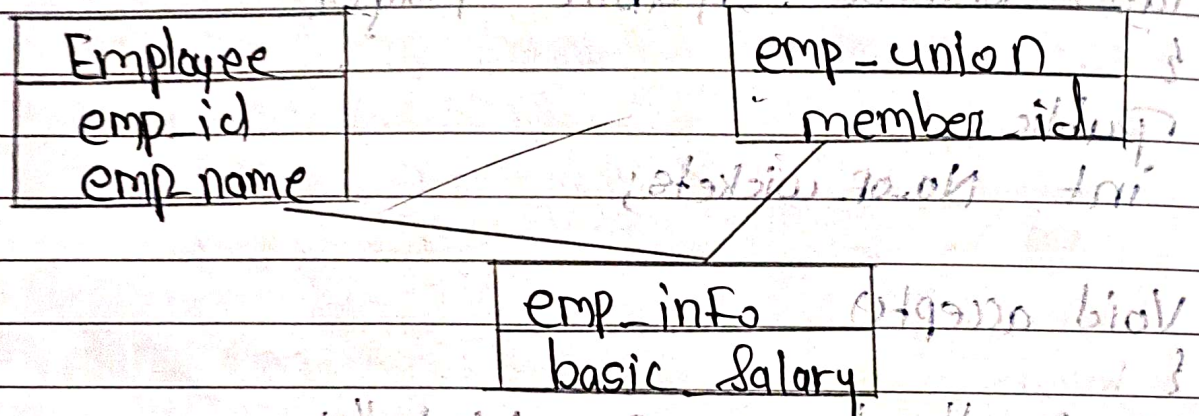
```
void main()
{
```

```
    Batsman b;
    b.read();
    b.show();
}
```

```

Bowler bl;
bl.acceptor;
bl.display();
getch();
}
    
```

Q.2 Write a program to show following Relationship using inheritance.



```

#include <iostream.h>
#include <conio.h>
class Employee
{
public:
int emp_id;
char emp_name[50];
};

class emp_union
{
public:
int member_id;
};
    
```



```
class emp_info : public Employee, public emp_union  
{
```

```
public:  
int basic_salary;
```

```
void accept()
```

```
{  
cout << "Enter emp_id, emp_name, member_id,  
and basic_salary:";  
cin >> emp_id >> emp_name >> member_id >>  
basic_salary;
```

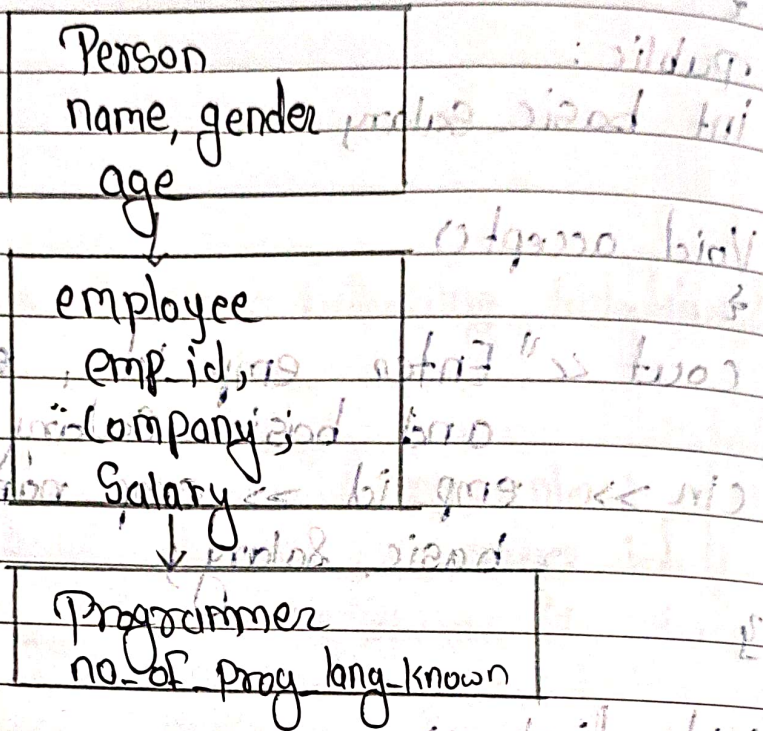
```
void display()
```

```
{  
cout << "Employee Details:" << emp_id << emp_name  
<< member_id << basic_salary;
```

```
void main()
```

```
{  
emp_info i;  
i.accept();  
i.display();  
getch();
```

Q.3. Write a program to define following Relationship



```

#include <iostream.h>
#include <conio.h>
class Person
{
    char name[50];
    char gender[20];
    int age;
};
  
```

```

class Employee : public Person
{
    public:
    int emp_id;
    int salary;
    char company[50];
};
  
```

```
class Programmer : public employee
```

```
{
```

```
public:
```

```
int no_of_prog_lang_known;
```

```
void accept()
```

```
{
```

```
cout << "Enter name, gender, age, emp_id,  
Salary, Company & no. of programming  
language known:";
```

```
cin >> name >> gender >> age >> emp_id >> salary  
>> company >> no_of_prog_lang_known;
```

```
}
```

```
void display()
```

```
{
```

```
cout << "Employee Details:" << name << gender << age  
<< emp_id << salary << company << no. of  
prog lang known;
```

```
}
```

```
}
```

```
void main()
```

```
{
```

```
Programmer P;
```

```
P.accept();
```

```
P.display();
```

```
getch();
```

```
}
```