Unit-3 (program solution)

1) WAP to enter information of n students and then display is using the concept multiple inheritance[PU: 2015 fall]

```
#include <iostream>
using namespace std;
class Student_info
{
private:
char name[25];
int age;
public:
void get_info()
{
cout<<"Enter name and age of student"<<endl;
cin>>name>>age;
}
void disp_info()
{
cout<<"Name of student:"<<name<<endl;
cout<<"Age of student:"<<age<<endl;
}
};
class Faculty
{
private:
char faculty[20];
public:
void get_faculty()
{
cout<<"Enter Faculty of student"<<endl;
cin>>faculty;
}
void disp_faculty()
{
cout<<"Faculty of student:"<<faculty<<endl;
}
};
```

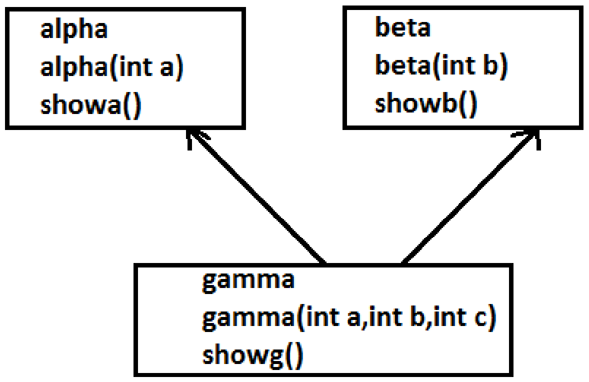
```
class Attendance:public Faculty,public Student_info
{
private:
int attendance;
public:
void get_attendance()
{
cout<<"Enter student's attendance percentage"<<endl;
cin>>attendance;
}
void disp_attendance()
{
cout<<"Student's attendance percentage ="<<attendance<<endl;
}
};

int main()
{
int i,n;
Attendance a[50];
cout<<"Enter Number of students"<<endl;
cin>>n;
for(i=0;i<n;i++)
{
cout<<"Enter information of student:"<<i+1<<endl;
a[i].get_info();
a[i].get_faculty();
a[i].get_attendance();
}
for(i=0;i<n;i++)
{
cout<<"Information of student:"<<i+1<<endl;
a[i].disp_info();
a[i].disp_faculty();
a[i].disp_attendance();
}
return 0;
}
```

2) WAP to enter information of n students and then display is using the concept multilevel inheritance.

```
#include <iostream>
using namespace std;
class Student_info
{
private:
char name[25];
int age;
public:
void get_info()
{
cout<<"Enter name and age of student"<<endl;
cin>>name>>age;
}
void disp_info()
{
cout<<"Name of student:"<<name<<endl;
cout<<"Age of student:"<<age<<endl;
}
};

class Faculty:public Student_info
{
private:
char faculty[20];
public:
void get_faculty()
{
cout<<"Enter Faculty of student"<<endl;
cin>>faculty;
}
void disp_faculty()
{
cout<<"Faculty of student:"<<faculty<<endl;
}
};
```

```
class Attendance:public Faculty
{
private:
int attendance;
public:
void get_attendance()
{
cout<<"Enter student's attendance percentage"<<endl;
cin>>attendance;
}
void disp_attendance()
{
cout<<"Student's attendance percentage ="<<attendance<<endl;
}
};

int main()
{
int i,n;
Attendance a[50];
cout<<"Enter Number of students"<<endl;
cin>>n;
for(i=0;i<n;i++)
{
cout<<"Enter information of student:"<<i+1<<endl;
a[i].get_info();
a[i].get_faculty();
a[i].get_attendance();
}
for(i=0;i<n;i++)
{
cout<<"Information of student:"<<i+1<<endl;
a[i].disp_info();
a[i].disp_faculty();
a[i].disp_attendance();
}
return 0;
}
```

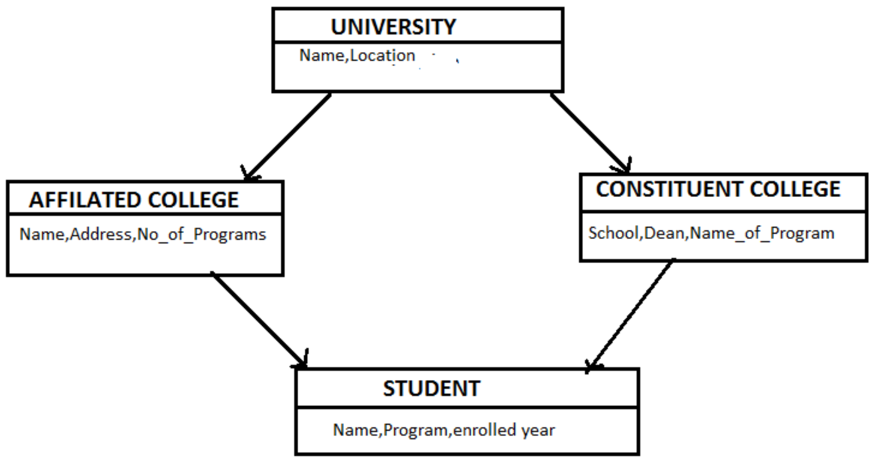
3) Write a complete program with reference to the given figure.



```
#include<iostream>
using namespace std;
class alpha
{
private:
int x;
public:
alpha(int a)
{
x=a;
}
void showa()
{
cout<<"x="<<x<<endl;
}
};

class beta
{
private:
int y;
public:
beta(int b)
{
y=b;
}
void showb()
{
cout<<"y="<<y<<endl;
}
};
```

```
class gamma:public beta,public alpha
{
private:
int z;
public:
gamma(int a,int b,int c):alpha(a),beta(b)
{
z=c;
}
void showg()
{
cout<<"z="<<z<<endl;
}
};

int main()
{
gamma g(5,10,15);
g.showa();
g.showb();
g.showg();
return 0;
}
```

4) The following figure shows the minimum information required for each class. Write a progam by realizing the necessary member functions to read and display information of individual object. Every class should contain at least one constructor and should be inherited from other classes as well.[PU:2019 fall]


UNIVERSITY
Name,Location
AFFILATED COLLEGE
Name,Address,No_of_Programs
CONSTITUENT COLLEGE
School,Dean,Name_of_Program
STUDENT
Name,Program,enrolled year

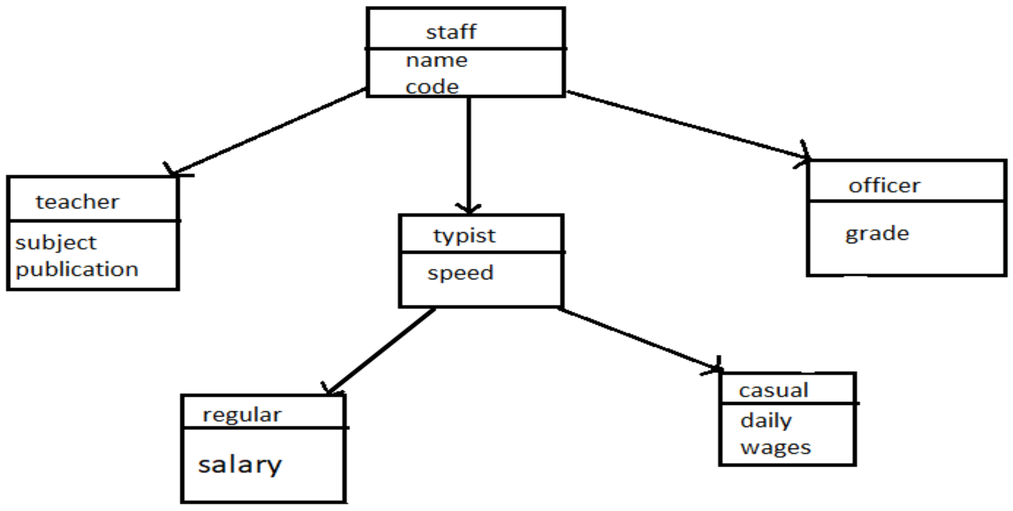
```
#include<iostream>
#include<string.h>
using namespace std;
```

```
class University
{
private:
char uname[20];
char location[20];
public:
University(char un[],char loc[])
{
strcpy(uname,un);
strcpy(location,loc);
}

void display()
{
cout<<"University name="<<uname<<endl;
cout<<"University location="<<location<<endl;
}
};

class Affiliated_College:virtual public University
{
private:
char cname[20];
char address[20];
int no_of_programs;
public:
Affiliated_College(char un[],char loc[],char cn[],char addr[],int nop):University(un,loc)
{
strcpy(cname,cn);
strcpy(address,addr);
no_of_programs=nop;
}
void display()
{
cout<<"Affiliated College Name="<<cname<<endl;
cout<<"Address="<<address<<endl;
cout<<"Number of programs="<<no_of_programs<<endl;
}
};
```

```
class Constituent_College:virtual public University
{
private:
char school[20];
char dean[20];
char name_of_program[20];
public:
Constituent_College(char un[],char loc[],char sn[],char d[],char npro[]):University(un,loc)
{
strcpy(school,sn);
strcpy(dean,d);
strcpy (name_of_program,npro);
}

void display()
{
cout<<"School name="<<school<<endl;
cout<<"Dean name="<<dean<<endl;
cout<<"Name of Program="<<name_of_program<<endl;
}
};

class Student:public Affiliated_College,public Constituent_College
{
private:
char student_name[20];
char program[20];
int enrolled_year;
public:
Student(char un[],char loc[],char cn[],char addr[],int nop, char sn[],
char d[], char npro[],char stn[],char pro[],int year):
Affiliated_College(un,loc,cn,addr,nop),Constituent_College(un,loc,sn,d,npro),University(un,loc)
{
strcpy(student_name,stn);
strcpy(program,pro);
enrolled_year=year;
}
void display()
{
cout<<"Student name="<<student_name<<endl;
cout<<"Program="<<program<<endl;
cout<<"Enrolled year="<<enrolled_year<<endl;
}
};
```

```
int main()
{
Student st("Pokhara","Dhungepatan","EEC","Sanepa",3,
"Engineering","Bharat","Civil","Pradip","Civil",2018);
st.University::display();
st.Affiliated_College::display();
st.Constituent_College::display();
st.display();
return 0;
}
```

5) An educational institution wishes to maintain a database of its employees. The database is divided into a number of classes whose hierarchical relationship are shown below. The figure also shows minimum information requires for each class. Specify all the classes and define functions to create database and retrieve individual information when required.



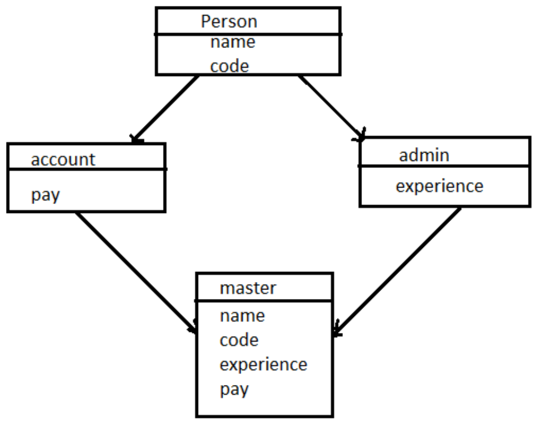
```
#include<iostream>
using namespace std;
class Staff
{
private:
char name[20];
int code;
public:
void get_staff()
{
cout<<"Enter Name and code of staff"<<endl;
cin>>name>>code;
}
void disp_staff()
{
cout<<"Name="<<name<<endl;
cout<<"Code="<<code<<endl;
}
};


class Teacher:public Staff
{
private:
char subject[20],publication[20];
public:
void get_teacher()
{
cout<<"Enter Subject and Publication"<<endl ;
cin>>subject>>publication;
}
void disp_teacher()
{
cout<<"Subject="<<subject<<endl;
cout<<"Publication="<<publication<<endl;
}
};

class Officer:public Staff
{
private:
char grade;
public:
void get_officer()
{
cout<<"Enter the grade"<<endl;
cin>>grade;
}
void disp_officer()
{
cout<<"Grade="<<grade<<endl;
}
};
```

```
class Typist:public Staff
{
private:
int speed;
public:
void get_typist()
{
cout<<"Enter Speed"<<endl;
cin>>speed;
}
void disp_typist()
{
cout<<"Speed="<<speed<<endl;
}
};

class Regular:public Typist
{
private:
int salary;
public:
void get_regular()
{
cout<<"Enter Salary"<<endl;
cin>>salary;
}
void disp_regular()
{
cout<<"Salary="<<salary<<endl;
}
};

class Casual:public Typist
{
private:
int wages;
public:
void get_casual()
{
cout<<"Enter Daily wages"<<endl;
cin>>wages;
}
void disp_casual()
{
cout<<"Daily wages="<<wages<<endl;
}
};
```

```
int main()
{
Teacher t;
Officer o;
Regular r;
Casual c;
t.get_staff();
t.get_teacher();
t.disp_staff();
t.disp_teacher();
o.get_staff();
o.get_officer();
o.disp_staff();
o.disp_officer();
r.get_staff();
r.get_typist();
r.get_regular();
r.disp_staff();
r.disp_typist();
r.disp_regular();
c.get_staff();
c.get_typist();
c.get_casual();
c.disp_staff();
c.disp_typist();
c.disp_casual();
return 0;
}
```

6) The following figure shows minimum information required for each class.

i) Write a Program to realize the above program with necessary member functions to create the database and retrieve individual information



```
#include<iostream>
using namespace std;
class Person
{
protected:
char name[20];
int code;
};

class Account:virtual public Person
{
protected:
float pay;
};

class Admin:virtual public Person
{
protected:
int experience;
};

class Master:public Account,public Admin
{
public:
void getinfo()
{
cout<<"Enter name of person"<<endl;
cin>>name;
cout<<"Enter code"<<endl;
cin>>code;
cout<<"Enter pay for person"<<endl;
cin>>pay;
cout<<"Enter experience"<<endl;
cin>>experience;
}
void display()
{
cout<<"Name:"<<name<<endl;
cout<<"Code:"<<code<<endl;
cout<<"Pay:"<<pay<<endl;
cout<<"Experience:"<<experience<<"years"<<endl;
}
};
```

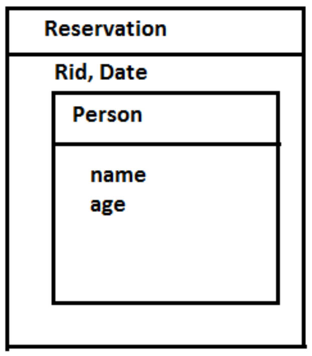
```
int main()
{
Master m;
m.getinfo();
m.display();
return 0;
}
```

ii) Rewrite the above program using constructor on each class to initialize the data members.

```
#include<iostream>
#include<string.h>
using namespace std;

class Person
{
protected:
char name[20];
int code;

public:
Person(char n[],int c)
{
strcpy(name,n);
code=c;
}
};

class Account:virtual public Person
{
protected:
float pay;
public:
Account(char n[],int c,float p):Person(n,c)
{
pay=p;
}
};
```

```
class Admin:virtual public Person
{
protected:
int experience;
Admin(char n[],int c,int e):Person(n,c)
{
experience=e;
}
};
class Master:public Account,public Admin
{
public:
Master(char n[],int c,float p,int e):Admin(n,c,e),Account(n,c,p),Person(n,c)
{
strcpy(name,n);
code=c;
pay=p;
experience=e;
}
void display()
{
cout<<"Name:"<<name<<endl;
cout<<"Code:"<<code<<endl;
cout<<"Pay:"<<pay<<endl;
cout<<"Experience:"<<experience<<"years"<<endl;
}
};
int main()
{
Master m("Pradip",121,23000.56,2);
m.display();
return 0;
}
```

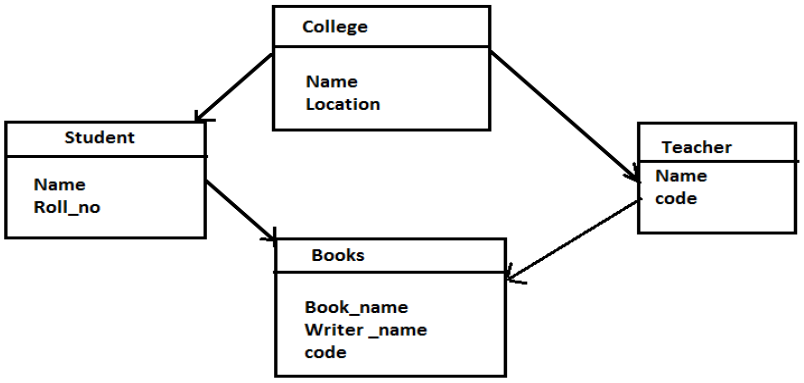
7. Write a program that allow you to book a ticket for person and use two classes Person, Reservation. Class Reservation is composite class/ container class.



```
#include<iostream>
using namespace std;
class Person
{
private:
char name[20];
int age;
public:
void getdata()
{
cout<<"Enter name and age of person"<<endl;
cin>>name>>age;
}
void display()
{
cout<<"Name="<<name<<endl;
cout<<"Age="<<age<<endl;
}
};

class Reservation
{
private:
int rid;
char date[10];
Person p;
public:
void getdata()
{
p.getdata();
cout<<"Enter Reservation ID"<<endl;
cin>>rid;
cout<<"Enter date in YY-MM-DD format "<<endl;
cin>>date;
}
void display()
{
p.display();
cout<<"Reservation ID="<<rid<<endl;
cout<<"Reservation date="<<date;
}
};
```

```
int main()
{
Reservation r;
r.getdata();
r.display();
return 0;
}
```

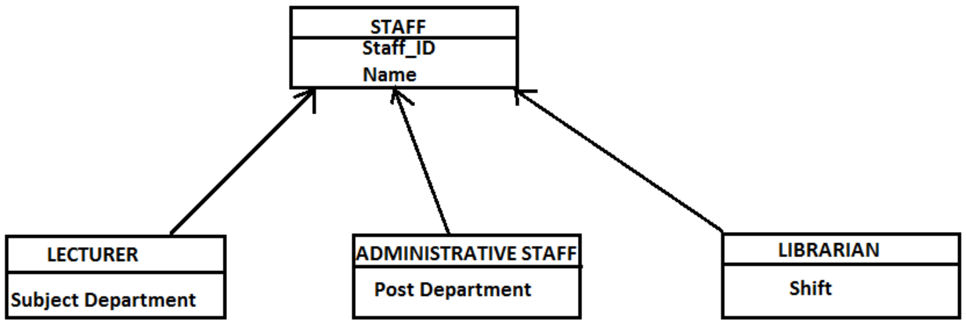
8. The following figure shows the minimum information required for each class. Write a program to realize the above program with necessary member functions to create the database and retrieve individual information .Every class should contain at least one constructor and should be inherited to other classes as well.[PU:2010 spring][PU 2009 fall]



```
#include<iostream>
#include<string.h>
using namespace std;
class College
{
private:
char cname[20];
char location[20];
public:
College(char cn[],char loc[])
{
strcpy(cname,cn);
strcpy(location,loc);
}
void display()
{
cout<<"College name="<<cname<<endl;
cout<<"College location="<<location<<endl;
}
};
```

```
class Student:virtual public College
{
private:
char sname[20];
int roll;
public:
Student(char cn[],char loc[],char sn[],int r):College(cn,loc)
{
strcpy(sname,sn);
roll=r;
}
void display()
{
cout<<"Student name="<<sname<<endl;
cout<<"Roll no="<<roll<<endl;
}
};

class Teacher:virtual public College
{
private:
char tname[20];
int tcode;
public:
Teacher(char cn[],char loc[],char tn[],int tc):College(cn,loc)
{
strcpy(tname,tn);
tcode=tc;
}
void display()
{
cout<<"Teacher name="<<tname<<endl;
cout<<"Teacher Code="<<tcode<<endl;
}
};
class Books:public Student,public Teacher
{
private:
char book_name[20];
char writer_name[20];
int book_code;
public:
Books(char cn[],char loc[],char sn[],int r, char tn[],int tc,
char bn[],char wn[],intbc):Teacher(cn,loc,tn,tc),Student(cn,loc,sn,r),College(cn,loc)
{
strcpy(book_name,bn);
strcpy(writer_name,wn);
book_code=bc;
}
void display()
{
cout<<"Book name="<<book_name<<endl;
cout<<"Writer name="<<writer_name<<endl;
cout<<"Book code="<<book_code<<endl;
} };
int main()
{
Books b("EEC","sanepa","Ram",47,"Pradip",151,"OOPs","Timothy Budd",53421);
b.College::display();
b.Student::display();
b.Teacher::display();
b.display();
return 0;
}
```

9) Develop a complete program for an institution, which wishes to maintain a database of its staff. The database is divided into number of classes whose hierarchical relationship is shown in the following diagram. specify all classes and define constructors and functions to create database and retrieve the individual information as per requirements.



```
#include<iostream>
#include<string.h>
using namespace std;


class STAFF
```

```
{
private:
int staff_id;
char name[20];
public:
STAFF(int sid,char sn[])
{
staff_id=sid;
strcpy(name,sn);
}
void display()
{
cout<<"Staff ID:"<<staff_id<<endl;
cout<<"Name:"<<name<<endl;
}
};

class LECTURER:public STAFF
{
private:
char subject[20];
char department[20];
public:
LECTURER(int sid,char sn[],char sub[],char dept[]):STAFF(sid,sn)
{
strcpy(subject,sub);
strcpy(department,dept);
}
void display()
{
cout<<"Subject:"<<subject<<endl;
cout<<"Department:"<<department<<endl;
}
};
class ADMINISTRATIVE_STAFF:public STAFF
{
private:
char post[20];
char department[20];
public:
ADMINISTRATIVE_STAFF(int sid,char sn[],char p[],char dept[]):STAFF(sid,sn)
{
strcpy(post,p);
strcpy(department,dept);
}

void display()
{
cout<<"Post:"<<post<<endl;
cout<<"Department:"<<department<<endl;
}
};

class LIBRARIAN:public STAFF
{
private:
char shift[20];
public:
LIBRARIAN(int sid,char sn[],char s[]):STAFF(sid,sn)
{
strcpy(shift,s);
}
void display()
{
cout<<"Shift:"<<shift<<endl;
}
};

int main()
{
LECTURER l(101,"Pradip Paudel","C programming","Computer");
l.STAFF::display();
l.display();
ADMINISTRATIVE_STAFF a(212,"Ekraj Acharaya","Admin","Administration");
a.STAFF::display();
a.display();
LIBRARIAN lib(314,"Sita sharma","Morning");
lib.STAFF::display();
lib.display();
return 0;
}
```

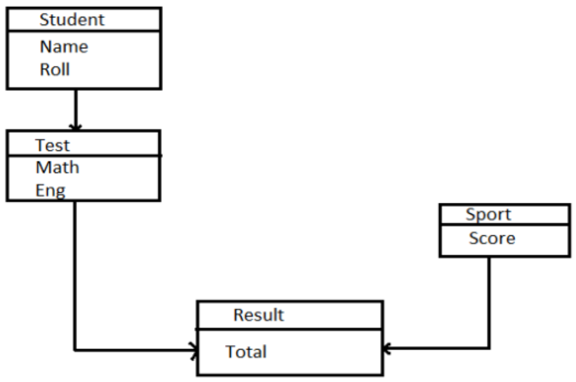
10) Develop a complete program for an institution which wishes to maintain a database of its staff. Declare a base class Staff which include staff_id and name.Now develop a records forthe following staffs with the given information below.
Lecturer(subject,department)
Administrative staff (Post,department)

```
#include<iostream>
#include<string.h>
using namespace std;


class Staff
```

```
{
private:
int staff_id;
char name[20];
public:
void get_staff()
{
cout<<"Enter staff id"<<endl;
cin>>staff_id;
cout<<"Enter name"<<endl;
cin>>name;
}
void display_staff()
{
cout<<"Staff ID:"<<staff_id<<endl;
cout<<"Name:"<<name<<endl;
}
};

class Lecturer:public Staff
{
char subject[20];
char department[20];
public:
void get_lecturer()
{
cout<<"Enter subject"<<endl;
cin>>subject;
cout<<"Enter department"<<endl;
cin>>department;
}
void display_lecturer()
{
cout<<"Subject:"<<subject<<endl;
cout<<"Department:"<<department<<endl;
}
};









class Administrative_staff:public Staff
```

```
{
char post[20];
char department[20];
public:
void get_administrative()
{
cout<<"Enter post"<<endl;
cin>>post;
cout<<"Enter department"<<endl;
cin>>department;
}
void display_administrative()
{
cout<<"Post:"<<post<<endl;
cout<<"Department:"<<department<<endl;
}
};
int main()
{
Lecturer l;
l.get_staff();
l.get_lecturer();
l.display_staff();
l.display_lecturer();
Administrative_staff a;
a.get_staff();
a.get_administrative();
a.display_staff();
a.display_administrative();
return 0;
}
```

12. Implement the below given in class diagram in C++.Assume necessary function yourself.



```
#include<iostream>
using namespace std;
class Student
{
private:
char name[20];
int roll;
public:
void get_student()
{
cout<<"Enter name and roll no"<<endl;
cin>>name>>roll;
}
void disp_student()
{
cout<<"Name="<<name<<endl;
cout<<"Rollno="<<roll<<endl;
}
};

class Test:public Student
{
protected:
float math,eng;
public:
void get_test()
{
cout<<"Enter marks in math and english"<<endl;
cin>>math>>eng;
}
void disp_test()
{
cout<<"marks in math="<<math<<endl;
cout<<"marks in english="<<eng<<endl;
}
};
class Sport
{
protected:
float score;
public:
void get_sport()
{
cout<<"Enter score"<<endl;
cin>>score;
}
void disp_sport()
{
cout<<"Score="<<score<<endl;
}
};

class Result:public Test,public Sport
{
private:
float total;
public:
void disp_total()
{
total=eng+math+score;
cout<<"Total="<<total<<endl;
}
};

int main()
{
Result r;
r.get_student();
r.get_test();
r.get_sport();
r.disp_student();
r.disp_test();
r.disp_sport();
r.disp_total();
return 0;
}
```

13. Create a class student with two data members represent name and roll. Use appropriate member function to read and print these data members name and roll. Derive a class marks from student that has additional data member sessional1, sessional2 to store sessional marks. Derive another class result from marks and add the sessional marks. Use appropriate member function to read and display data in the class.

```
#include<iostream>
using namespace std;
class Student
{
protected:
char name[20];
int roll;
public:
void getdata()
{
cout<<"Enter the name"<<endl;
cin>>name;
cout<<"Enter the rollno"<<endl;
cin>>roll;
}
void putdata()
{
cout<<"Name="<<name<<endl;
cout<<"Roll="<<roll<<endl;
}
};
class Marks:public Student
{
protected:
float sessional1;
float sessional2;public:
void getmarks()
{
cout<<"Enter sessional1 marks"<<endl;
cin>>sessional1;
cout<<"Enter sessional2 marks"<<endl;
cin>>sessional2;
}
void putmarks()
{
cout<<"sessional1="<<sessional1<<endl;
cout<<"sessional2="<<sessional2<<endl;
}
};
class Result:public Marks
{
private:
float total;
public:
void display()
{
total=sessional1+sessional2;
cout<<"Total marks="<<total<<endl;
}
};
```

```
int main()
{
Result r;
r.getdata();
r.getmarks();
r.putdata();
r.putmarks();
r.display();
return 0;
}
```

14. Create a class student with two data members to represent name and age. Use member function to read and print those data. From this derive a class called boarder with member data to represent room number .Derive another class called day-scholar from class student with member data to represent address of student. In both derived class use function to read and print respective data.

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

class Student
{
protected:
char name[20];
int age;
public:
void read()
{
cout<<"Enter name"<<endl;
cin>>name;
cout<<"Enter age"<<endl;
cin>>age;
}
void print()
{
cout<<"Name="<<name<<endl;
cout<<"Age="<<age<<endl;
}
};



class Boarder:public Student
```

```
{
private:
int roomnumber;
public:
void getroomno()
{
cout<<"Enter room number"<<endl;
cin>>roomnumber;
}
void putroomno()
{
cout<<"Room number="<<roomnumber<<endl;
}
};
class Day_scholar:public Student
{
private:
char address[20];
public:
void getaddress()
{
cout<<"Enter address"<<endl;
cin>>address;
}
void putaddress()
{
cout<<"Address="<<address<<endl;
}
};
int main()
{
Boarder b;
Day_scholar d;
b.read();
b.getroomno();
d.read();
d.getaddress();
b.print();
b.putroomno();
d.print();
d.putaddress();
return 0;    }
```