Access specifiers

Multiple inheritance : when a derived class is formed by inheriting the properties of 2 or more base classes;

Student marks

Result

Class student

{

Protected :

Int rollno;

Char name[20];

Public:

Void getstud() ;

Void dispstud();

};

Class marks

{

Protected :

Int cs, maths, English;

Public:

Void getmarks();

Void dispmarks();

};

Class result : public student, public marks

{

Int total,per;

Public:

Void calc()

{

Total= cs+maths+english;

}

Void display()

{

Cout<<total;

}

};

Void main()

{

Result r;

r.getstud();

r.getmarks();

r.calc();

r.disstud();

r.dismarks();

r.display();

}

Private derivation

Class student

{

Protected :

Int rollno;

Char name[20];

Public:

Void getstud() ;

Void dispstud();

};

Class marks

{

Protected :

Int cs, maths, English;

Public:

Void getmarks();

Void dispmarks();

};

Class result : private student, private marks

{

Int total,per;

Public:

Void calc()

{

Getstud();

Getmarks();

Total= cs+maths+english;

}

Void display()

{

Disstud();

Dismarks();

Cout<<total;

}

};

Void main()

{

Result r;

r.calc();

r.display();

}

Using constructors in multiple inheritance

Class student

{

Protected :

Int rollno;

Char name[20];

Public:

Student(int r, char n[20])

{

Rollno = r;

Strcpy(name,n);

}

Void getstud() ;

Void dispstud();

};

Class marks

{

Protected :

Int cs, maths, English;

Public:

Marks(int c, int m, int e)

{

Cs= c;

Maths=m;

English = e;

}

Void getmarks();

Void dispmarks();

};

Class result : public student, public marks

{

Int total,per;

Public:

**Result(int r, char n[20, int c, int m, int e) : student(r,n ), marks(c,m,e)**

**{ }**

Void calc()

{

Total= cs+maths+english;

}

Void display()

{

Disstud();

Dismarks();

Cout<<total;

}

};

Void main()

{

Result r(1, “A”, 23, 25, 23);

r.calc();

r.display();

}