

### 3. Extending classes using inheritance

**Defination**

The mechanism of deriving a class from another class is known as inheritance.

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

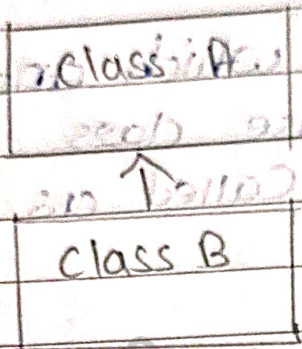
**\* Visibility modes & effects :**

Base Class member Visibility	Derived class Visibility		
	Public Derivation	protected Derivation	private Derivation
Public member	Public	protected	private
protected member	protected	protected	private
private member	Not inherited	Not inherited	not inherited

←  
←

### \* Types of inheritance

#### 1) Single inheritance



In this inheritance only one class is derived from another class

Syntax:

class derived class : derivation type base class name

Ex: `Class B : public A`

Q1) Write a program add two numbers using single inheritance such that the base class function must accept two number from user & derived class function must add this numbers and display the sum.

```

-> #include <stdio.h>
#include <conio.h>
class Data
{
public:
int a, b;

```

```

void read() {
    cout << "Enter a & b: ";
    cin >> a >> b;
}

```

```

class Sum : public Data {
public:
    int addition;
    void add() {
        addition = a + b;
    }
    void display() {
        cout << "sum: " << addition;
    }
}

```

```

void main() {
    Sum s;
    s.read();
    s.add();
    s.display();
}

```

Q2] Write a program to find the area of circle using single inheritance such that the base class function must accept the radius from user and derived class function must calculate & display the area.

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

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

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

```
class circle
```

```
{
```

```
public:
```

```
int r;
```

```
void read()
```

```
{
```

```
cout << "Enter radius:";
```

```
cin >> r;
```

```
}
```

```
};
```

```
class area : public circle
```

```
{
```

```
public:
```

```
int a;
```

```
void compute()
```

```
{
```

```
a = 3.14 * r * r;
```

```
}
```

```
void display()
```

```
{
```

```
cout << "area of circle:" << a;
```

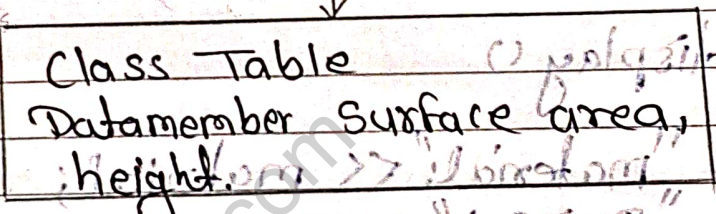
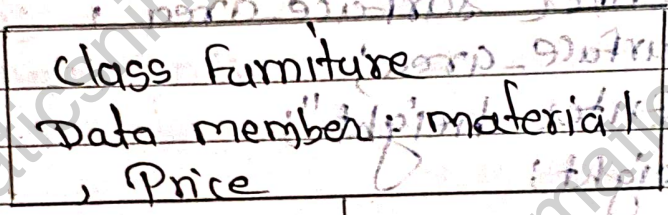
```
}
```

```
};
```

```

Void main()
{
    Area a1;
    a1.read();
    a1.compute();
    a1.display();
}
    
```

Q3. Write a program to implement single inheritance from following figure - accept & display the data for one table.



```

-> #include <iostream.h>
#include <conio.h>
class Furniture
{
public:
    char materia[50];
    float price;
};
    
```

```

class Table : public Furniture
{
public:
float surface-area, height;

```

```

void accept()

```

```

{
cout << "Enter material:";

```

```

cin >> material;

```

```

cout << "Enter price:";

```

```

cin >> price;

```

```

cout << "Enter surface-area";

```

```

cin >> surface-area;

```

```

cout << "Enter height";

```

```

cin >> height;
}

```

```

void display()

```

```

{
cout << "material:" << material;

```

```

cout << "price:" << price;

```

```

cout << "surface-area:" << surface-area;

```

```

cout << "height:" << height;
}

```

```

};

```

```

void main()

```

```

{

```

```

table t;

```

```

t.accept();

```

```

t.display();
}

```

Class Table : public furniture

```
{  
Public :  
float surface_area, height;
```

```
void accept()
```

```
{  
cout << "Enter material:" ;
```

```
cin >> material;
```

```
cout << "Enter price:" ;
```

```
cin >> price;
```

```
cout << "Enter surface_area";
```

```
cin >> surface_area;
```

```
cout << "Enter height:" ;
```

```
cin >> height;
```

```
}
```

```
void display()
```

```
{  
cout << "material:" << material;
```

```
cout << "price:" << price;
```

```
cout << "surface_area:" << surface_area;
```

```
cout << "height:" << height;
```

```
}
```

```
};
```

```
void main()
```

```
{
```

```
table t;
```

```
t.accept();
```

```
t.display();
```

```
}
```

Q4 Write a program to implement single inheritance declare base class as employee with emp\_no, emp\_name, declare derived class as fitness with height & weight accept & display data for one employee.

```
→ #include <iostream.h>
#include <conio.h>
class employee
{
public:
int emp_no;
char emp_name[50];
};

class fitness : public employee
{
public:
int height, weight;
void accept()
{
cout << "Enter emp_name & emp_no:";
cin >> emp_name >> emp_no;
cout << "Enter height & weight:";
cin >> height >> weight;
}

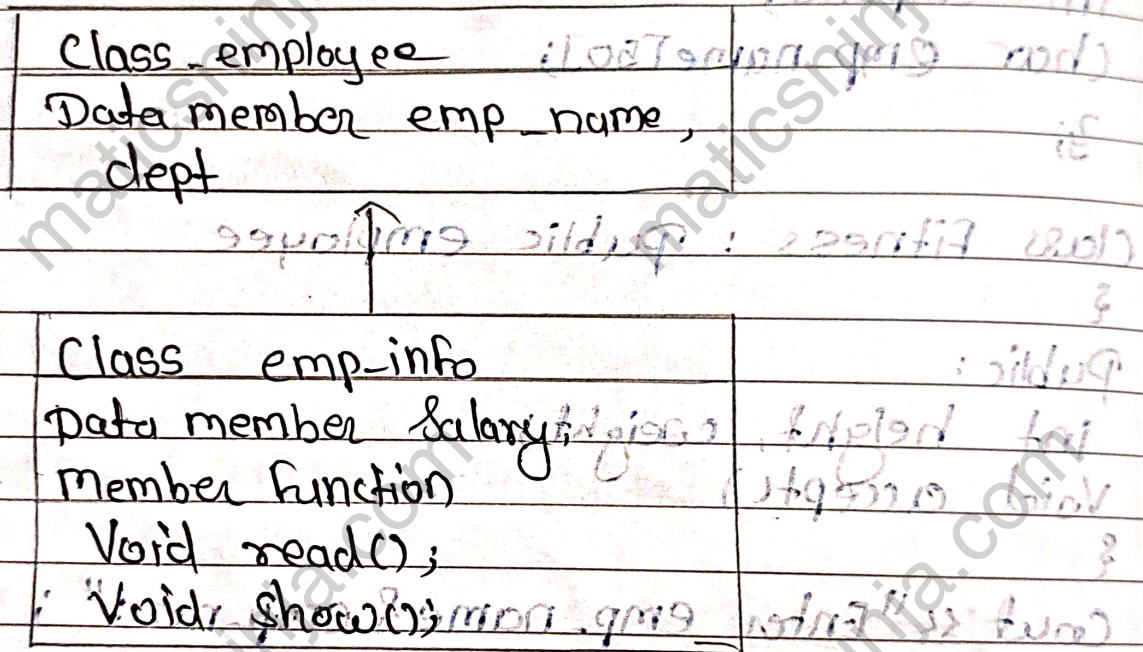
void display()
{
cout << "Employee Details:" << emp_name <<
emp_no << height << weight;
}
}
```



```

};
void main()
{
    Fitness f;
    f.accept();
    f.display();
}
    
```

Q.5 Write a program to single inheritance from the following figure.



```

#include <iostream.h>
#include <conio.h>
class employee
{
public:
    char emp_name[50], dept[50];
};
    
```

```
class emp_info : public employee
{
public:
float salary;

void read()
{
cout << "Enter emp_name and dept_name & salary:"
cin >> emp_name >> dept >> salary;
}

void show()
{
cout << "Employee Details:" << emp_name << dept
<< salary;
}
};

void main()
{
emp_info e;
e.read();
e.show();
}
```

Q6. Write a program to implement single inheritance from following Fig. accept data for 8 accounts and display details of account having balance < 10,000

```

Class Bank
account no;
    
```

```

Class Account
acc_name;
balance;
    
```

→

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

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

```
class Bank
```

```
{
```

```
public:
int account no;
```

```
};
```

```
class account : public Bank
```

```
{
```

```
public:
```

```
char acc_name[50];
```

```
int balance;
```

```
void accept()
```

```
{
```

```
cout << "Enter account number";
```

```
cin >> account no;
```

```
cout << "Enter account name:";
```

```
cin >> acc_name;
```

```
cout << "Enter balance:";
```

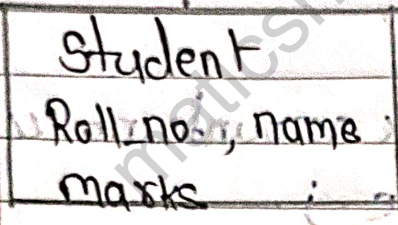
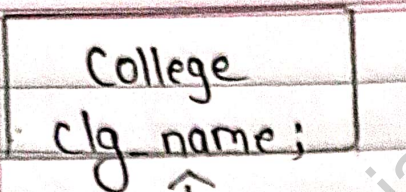
```
cin >> balance;
```

↓

```
void display()
{
    if (balance < 10,000)
    {
        cout << "Account details : " << account no << endl;
        ac-name << balance ;
    }
}
```

```
void main()
{
    account a[8];
    int i;
    for (i=0; i<8; i++)
    {
        a[i].accept();
    }
    for (i=0; i<8; i++)
    {
        a[i].display();
    }
}
```

Q7. write a program: Single inheritance from following fig & display the student data (accept data for 10 student) and display name of student whose marks are > 70.



```
→ #include <iostream.h>
#include <conio.h>
class College
{
public:
char clg_name [50];
};

class Student : public College
{
public:
int Roll-no, marks;
char name [50];

void accept ()
{
cout << "Enter college name:";
cin >> clg_name;
cout << "Enter Roll no:";
cin >> Roll-no;
cout << "marks:";
cin >> marks;
cout << "Enter name:";
cin >> name;
}
```

```
void display ()
```

```
{
```

```
    if (marks > 70)
```

```
        cout << "Student Details: " << clg_name << " Roll-no " << name << marks ;
```

```
}
```

```
}
```

```
}
```

```
void main ()
```

```
{
```

```
    student s[10];
```

```
    int i;
```

```
    for (i=0; i<10; i++)
```

```
    {
```

```
        s[i].accept();
```

```
    } for (i=0; i<10; i++)
```

```
    {
```

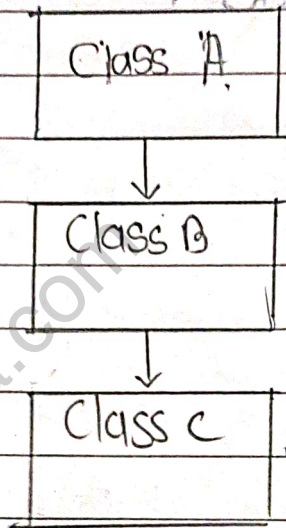
```
        s[i].display();
```

```
    }
```

```
}
```

## 2] Multilevel inheritance

In this case one class is derive from a class which is also derive from another class.



Syntax:

```
Class A  
{  
  ;  
};  
Class B : Public A  
{  
  ;  
};  
Class C : Public B  
{  
  ;  
};
```

Here Class C is derive from Class B which is derived from Class A.

Q1. Write a program to calculate percentage of a student using multi level inheritance. The base class function will accept the marks in 3 Subject from user. A class will be derived from the above mention class that will have a function to find the total marks obtained & another class derive from these will have functions to calculate & display the percentage score.

```
→ #include <iostream.h>
#include <conio.h>
class student
{
public:
int m1, m2, m3;
void read()
{
cout << "Enter 3 sub marks:";
cin >> m1 >> m2 >> m3;
}
}

class sum : public student
{
public:
int total;
void addition()
{
total = m1 + m2 + m3;
}
}
```



class Percentage { public sum

public: float p;

void calculate (int a, int b, int c)

{  

$$p = \frac{(a+b+c)}{3} \times 100;$$
}

void display ()

{ cout << "percentage = " << p ;

void main ()

{ Percentage m;

m.read();

m.addition();

m.calculation();

m.display();

Q2. Write a program to calculate the volume of a sphere using multilevel inheritance. The base class O function will accept the radius from user. The class will be derived from above mentioned class. That will have a function to find the area of a circle.

and another class derived from these we have functions to calculate & display the volume of the sphere.

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

```
{
public:
int r, a, v;
void accept()
{
cout << "Enter r";
cin >> r;
};
};
```

```
class area : public sphere
```

```
{
public:
void areacircle()
{
a = 3.14 * r * r;
};
};
```

```
class result : public area
```

```
{
public:
void calculate()
{
```

radius  
Volume

radius  
area

```

V = (4/3) * pi * r * r * r;
}

void display()
{
    cout << "Volume of Sphere: " << V;
}

void main()
{
    result r;
    r.accept();
    r.areaofcircle();
    r.calculate();
    r.display();
}
    
```

Q3. Identify the type of inheritance & implement it by writing the program for following Fig. Assume suitable member()

Customer  
name;  
Phone no;

Depositor  
Acc no;  
balance;

Borrower  
Loan\_no;  
loan\_amount;

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

```
class Customer  
{
```

```
public:  
char name[50];  
int phone_no;  
};
```

```
class Depositor : public Customer  
{
```

```
public:  
int Acc_no, balance;  
};
```

```
class Borrower : public Depositor  
{
```

```
public:  
int loan_no;  
float loan_amount;
```

```
void accept()  
{
```

```
cout << "name";  
cin >> name;  
cout << "Enter phone_no:";  
cin >> phone_no;
```

```
Cout << "Enter Account no:";
```

```
Cin >> ACC_no;
```

```
Cout << "Enter balance:";
```

```
Cin >> balance;
```

```
Cout << "Enter loan no:";
```

```
Cin >> loan_no;
```

```
Cout << "Enter loan amount:";
```

```
Cin >> loan_amount;
```

```
}
```

```
Void display ()
```

```
{
```

```
Cout << "Name:" << name;
```

```
Cout << "Phone no:" << phone_no << endl;
```

```
Cout << "Account no:" << ACC_no << endl;
```

```
Cout << "Balance:" << balance << endl;
```

```
Cout << "loan-no:" << loan_no << endl;
```

```
Cout << "loan amount:" << loan_amount;
```

```
}
```

```
};
```

```
Void main ()
```

```
{
```

```
borrower b;
```

```
b.accept();
```

```
b.display();
```

```
}
```

Q4. Write a C++ program to declare a class a college with members as clg-code derive a new class a student with members as stud-id accept & display details of student along with college for 1 object of student.

```
→ #include <iostream.h>
#include <conio.h>
class college
{
public:
int clg_code;
};

class student : public college
{
public:
int student_id;
void accept()
{
cout << "Enter clg_code : ";
cin >> clg_code;
cout << "Enter Student id : ";
cin >> student_id;
}
void display()
{
cout << "The details are : " << clg_code << student_id;
}
};
```



```
Cout << "Enter Account no:";
```

```
Cin >> ACC no;
```

```
Cout << "Enter balance:";
```

```
Cin >> balance;
```

```
Cout << "Enter loan no:";
```

```
Cin >> loan no;
```

```
Cout << "Enter loan amount:";
```

```
Cin >> loan amount;
```

```
}
```

```
Void display ()
```

```
{
```

```
Cout << "Name:" << name;
```

```
Cout << "Phone no:" << phone no << endl;
```

```
Cout << "Account no:" << ACC no << endl;
```

```
Cout << "Balance:" << balance << endl;
```

```
Cout << "loan no:" << loan no << endl;
```

```
Cout << "loan amount:" << loan amount;
```

```
}
```

```
};
```

```
Void main ()
```

```
{
```

```
borrower b;
```

```
b. accept();
```

```
b. display();
```

```
}
```

Q4. Write a C++ program to declare a class a college with members as clg\_code derive a new class a student with members as stud\_id accept & display details of student along with college for 1 object of student

```
→ #include <iostream.h>
#include <conio.h>
class college
{
public:
int clg_code;
};

class student : public college
{
public:
int student_id;
void accept()
{
cout << "Enter clg code : ";
cin >> clg_code;
cout << "Enter student id : ";
cin >> student_id;
}
void display()
{
cout << "The details are : " << clg_code << " student_id : ";
}
};
```



```

void main()
{
    Student s;
    s.accept();
    s.display();
}
    
```

Q5 Write a program to declare a class employee with member as emp name & emp id derive a new class department with member as resignation again derive a new class as employee info with members as Salary accept & display of this data for five employee.

```

-> #include <iostream.h>
#include <conio.h>
class employee
{
public:
    char emp_name[50];
    int emp_id;
};
class department : public employee
{
    char resignation[50];
};
class info : public department
{
public:
    float salary;
}
    
```

```
void accept()
{
    cout << "Enter employee name :";
    cin >> emp_name;
    cout << "Enter employee id :";
    cin >> emp_id;
    cout << "Enter resignation :";
    cin >> resignation;
    cout << "Enter Salary :";
    cin >> Salary;
}
```

```
void display()
{
    cout << "Employee name : " << emp_name;
    cout << "Employee id : " << emp_id;
    cout << "Resignation : " << resignation;
    cout << "Salary : " << Salary;
}
```

```
void main()
{
    int i;
    info i1[5];
    for (i=0; i<5; i++)
    {
        i1[i].accept();
    }
    for (i=0; i<5; i++)
    {
        i1[i].display();
    }
}
```