Inheritance

Date / /
Page

Inheritance is the important features of object oriented programming language. It is a ~~mechanism~~ mechanism in which a new class (derived class) is derived from an existing class or an old class. ~~(The new derived class is also called)~~ An existing old class is also called parrent or Super class (base class). In Inheritance one or more classes are linked with another one or more classes. The derived class can access protected data members and member function of a base (super) class directly. It cannot access private data member or member function directly.

Types of Inheritance

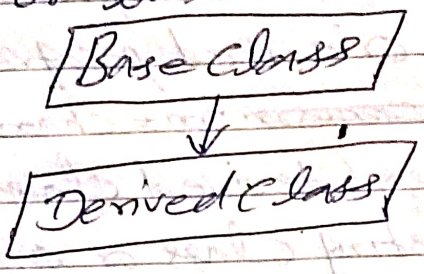
There are following types of Inheritance

i) Single Inheritance.
ii) Multiple Inheritance.
iii) Multilevel Inheritance.
iv) Hierarchical Inheritance.
v) Hybrid Inheritance.

i) Single Inheritance :- In this type of inheritance, single derived class is derived from a single super

Date / /
Page

Class or base class.



When a class inherits from a Single base class, it is known a Single inheritance.

For e.g. →

```
#include<iostream.h>
#include<conio.h>
Class Worker
{
 int age;
 char name[10];
 public:
  Void get();
  Void show();
 };
Void Worker::get()
 {
  Cout<<"\n Your name please";
  Cin>>name;
  Cout<<"\n Your age please";
  Cin>>age;
 }
Void worker::show()
 {
```

```
cout<<"\n My name is:"<<name<<"\n
    My age is:"<<age;
}
Class Manager: public worker          // derived publically
{
int now;
public:
void get();
void show();
};
Void manager::get()
{
worker::get(); //the Calling of base class input fn.
cout<<"\n Number of workers under you";
cin>>now;
}
Void manager::show()
{
worker::show();
cout<<"\n No. of workers"<<now;
}
Void main()
{
clrscr();
worker W1;
Manager M1;
M1.get();
M1.show();
}
```