Inheritance

Inheritance: Creating new class from existing class such that the features of existing class are available to the new class is called inheritance. Already existing class is called super class & produced class is called sub class. Using inheritance while creating sub classes a programmer can reuse the super class code without rewriting it.

Program 1: Write a program to create a Person class which contains general details of a person and create a sub class Employ which contains company details of a person. Reuse the general details of the person in its sub class.

```
// Inheritance Example
class Person
{       String name;
        String permanentAddress;
        int age;
        void set_PermanentDetails (String name, String permanentAddress, int age)
        {       this.name = name;
                this.permanentAddress = permanentAddress;
                this.age = age;
        }
        void get_PermanentDetails ()
        {       System.out.println ("Name : " + name);
                System.out.println ("Permanent Address : " + permanentAddress);
                System.out.println ("Age :" + age);
        }
}
class Employ extends Person
{       int id;
        String companyName;
        String companyAddress;
        Employ (int id, String name, String permanentAddress, int age,
                                        String companyName, String companyAddress)
        {       this.id = id;
                set_PermanentDetails (name, permanentAddress, age);
                this.companyName = companyName;
                this.companyAddress = companyAddress;
        }
        void get_EmployDetails ()
        {       System.out.println ("Employ Id : " + id);
                get_PermanentDetails ();
                System.out.println ("Company Name : "+ companyName);
                System.out.println ("Company Address : "+companyAddress);
        }
```

```
}
class InherDemo
{       public static void main (String args [])
        {       Employ e1 = new Employ (101, "Suresh Kumar", "18-Madhura Nagar-Tirupati",
                                        29, "Centris Software- Chennai", "20-RVS Nagar");
                e1.get_EmployDetails ();
        }
}
```

Program 2: Write a program to illustrate the order of calling of default constructor in super and sub class.

```
// Default constructors in super and sub class
class One
{       One ( )         //super class default constructor
        {
                System.out.println ("Super class default constructor called");
        }
}
class Two extends One
{       Two ( )         //sub class default constructor
        {
                System.out.println ("Sub class default constructor called");
        }
}
class Const
{       public static void main (String args[])
        {       Two t=new Two ( );   //create sub class object
        }
}
```

Super class default constructor is available to sub class by default.

First super class default constructor is executed then sub class default constructor is executed.

Super class parameterized constructor is not automatically available to subclass. super is the key word that refers to super class.

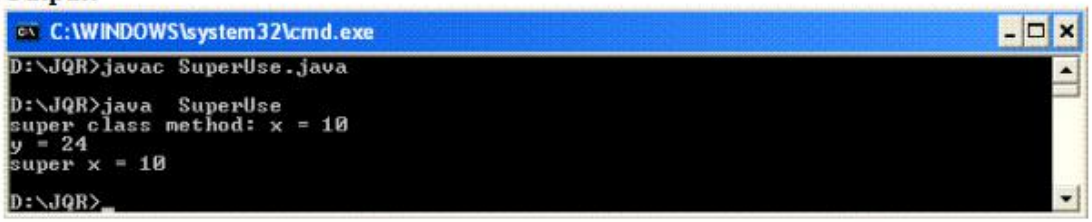
The keyword 'super':

- super can be used to refer super class variables as: super.variable
- super can be used to refer super class methods as: super.method ()
- super can be used to refer super class constructor as: super (values)

Program 3: Write a program to access the super class method, super class parameterized constructor and super class instance variable by using super keyword from sub class. // super refers to super class- constructors, instance variables and methods class A

```
class A
{       int x;
        A (int x)
        {
                this.x = x;
        }
        void show( )
        {       System.out.println("super class method: x = "+x);
        }
}
class B extends A
{       int y;
        B (int a,int b)
        {
                super(a);       // (or) x=a;
                y=b;
        }
        void show( )
        {       super.show ();
                System.out.println ("y = "+y);
                System.out.println (" super x = " + super.x);
        }
}
class SuperUse
{
        public static void main(String args[])
        {       B ob = new B (10, 24);
                ob.show ( );
        }
}
```

Output:



· Super key word is used in sub class only.

· The statement calling super class constructor should be the first one in sub class constructor