USING SUPER

The super keyword **refers to immediate parent class object**. Whenever you create the in- stance of subclass, an instance of parent class is created implicitly which is referred by superreference variable.

- It an be used to refer immediate parent class instance variable when both parent and child class have member with same name
- It can be used to invoke immediate parent class method when child class has overridden that method.
- super() can be used to invoke immediate parent class constructor.

Use of super with variables:

When both parent and child class have member with same name, we can use super key-word to access member of parent class.

Example:

```
class SuperCls
{
   int x = 20;
}
 /* sub class SubCls extending SuperCls */
class SubCls extends SuperCls
{
   int x = 80;
    void display()
   {
      System.out.println("Super Class x: " + super.x); //print x of super class
      System.out.println("Sub Class x: " + x);  //print x of subclass
   }
}
 /* Driver program to test
*/class Main
{
   public static void main(String[] args)
   {
      SubCls obj = new
      SubCls();obj.display();
   }
}
```

Sample Output:

```
Super Class x: 20

Sub Class x: 80
```

In the above example, both base class and subclass have a member x. We could access x of base class in sublcass using super keyword.

Use of super with methods:

The super keyword can also be used to invoke parent class method. It should be used ifsubclass contains the same method as parent class (Method Overriding).

```
class SuperCls
{
   int x = 20;
   void display(){ //display() in super class
      System.out.println("Super Class x: " + x);
   }
}
 /* sub class SubCls extending SuperCls */
class SubCls extends SuperCls
{
   int x = 80;
    void display()    //display() redefined in sub class – method overriding
   {
      System.out.println("Sub Class x: " + x);
      super.display();         // invoke super class
      display()
   }
}
 /* Driver program to test
*/class Main
{
   public static void main(String[] args)
   {
      SubCls obj = new
      SubCls();obj.display();
   }
}
```

Sample Output:

```
Sub Class x: 80
Super Class x: 20
```

In the above example, if we only call method display() then, the display() of sub class gets invoked. But with the use of super keyword, display() of superclass could also be invoked.

Use of super with constructors:

The super keyword can also be used to invoke the parent class constructor.

Syntax:

```
super();
```

- super() if present, must always be the first statement executed inside a subclass constructor.
- When we invoke a super() statement from within a subclass constructor, we are invoking the immediate super class constructor

Example:

```
class SuperCls
{
   SuperCls(){
      System.out.println("In Super Constructor");
   }
}
 /* sub class SubCls extending SuperCls */
class SubCls extends SuperCls
{
      SubCls(){
        super();
      System.out.println("In Sub Constructor");
   }
}
 /* Driver program to test */
class Main
{
   public static void main(String[] args)
   {
      SubCls obj = new SubCls();
   }
}
```

Sample Output:

```
In Super ConstructorIn
```

Sub Constructor

Invoking Superclass Parameterized Constructor

To call parameterized constructor of superclass, we must use the super keyword as shown below.

Syntax:

```
super(value);
```

Example:

```
class  SuperCls{ int
   x; SuperCls(int
   x){
      this.x=x;                // this refers to current invoking object
   }
}
class SubCls extends SuperCls{
   int y;
   SubCls(int x,int y){
      super(x);                // invoking parameterized constructor of superclass
      this.y=y;
   }
   public void display(){ System.out.println("x:
      "+x+" y: "+y);
   }
}
public class Main
{
     public static void main(String[] args) {
        SubCls obj=new SubCls(10,20);
        obj.display();
     }
}
```

Sample Output:

```
x: 10 y: 20
```

The program contains a superclass and a subclass, where the superclass contains a parameterized constructor which accepts a integer value, and we used the super keyword to invokethe parameterized constructor of the superclass.