Inherited method

- The method which would come from parent to child due to inheritance is called inherited method.

Example

```
class Parent{
        public void methodOne(){System.out.println("methodOne from parent");}
}
class Child extends Parent{
          public void methodTwo(){System.out.println("methodTwo from child");}
}

public class TestApp{
       public static void main(String... args){
                Parent p=new Parent();
               p.methodOne();

                Child c=new Child();
               c.methodOne();//inherited method
                c.methodTwo();//Specialized method

                Parent p1=new Child();
                p1.methodOne();
                p1.methodTwo();//CE: can't find the symbol methodTwo in Parent
       }
    }
```

Overridden Method

The method which is taken from Parent and changes the implementation as per the needs of the requirement in the class is called the “overridden method”.

Example

```
class Parent{
    public void methodOne(){System.out.println("methodOne from parent");}
}
class Child extends Parent{
        @Override
        public void methodOne(){System.out.println("methodOne from child");}
 }
 public class TestApp{
        public static void main(String... args){
              Parent p=new Parent();
               p.methodOne();//methodOne from parent

             Child c=new Child();
              c.methodOne();//methodOne from child
        }
   }
```

Inherited method

- The method which would come from parent to child due to inheritance is called inherited method.

Example

```
class Parent{
        public void methodOne(){System.out.println("methodOne from parent");}
}
class Child extends Parent{
          public void methodTwo(){System.out.println("methodTwo from child");}
}

public class TestApp{
       public static void main(String... args){
                Parent p=new Parent();
               p.methodOne();

                Child c=new Child();
               c.methodOne();//inherited method
                c.methodTwo();//Specialized method

                Parent p1=new Child();
                p1.methodOne();
                p1.methodTwo();//CE: can't find the symbol methodTwo in Parent
       }
    }
```

Overridden Method

The method which is taken from Parent and changes the implementation as per the needs of the requirement in the class is called the “overridden method”.

Example

```
class Parent{
    public void methodOne(){System.out.println("methodOne from parent");}
}
class Child extends Parent{
        @Override
        public void methodOne(){System.out.println("methodOne from child");}
 }
 public class TestApp{
        public static void main(String... args){
              Parent p=new Parent();
               p.methodOne();//methodOne from parent

             Child c=new Child();
              c.methodOne();//methodOne from child
        }
   }
```

Specialized Method

The methods which are specific to the particular class are called “Specialised method”.

Example

```
class Parent{
        public void methodOne(){System.out.println("methodOne from parent");}
}
class Child extends Parent{
       public void methodOne(){System.out.println("methodOne from child");}
       public void methodTwo(){System.out.println("methodTwo from child");}
}
public class TestApp{
       public static void main(String... args){
                Parent p=new Parent();
                 p.methodOne();//methodOne from parent

                 Child c=new Child();
                   c.methodOne();//methodOne from child
                   c.methodTwo();//methodOne from child

                 Parent p1=new Child();
                  p1.methodOne();
                  p1.methodTwo();//CE: can't find the symbol methodTwo in Parent
       }
}
```

Rules to Override a method

- Whatever the Parent has by default available to the Child through inheritance, if the Child is not satisfied with Parent class method implementation then Child is allowed to redefine that Parent class method in Child class in its own way this process is called overriding.
- The Parent class method which is overridden is called the overridden method.
- The Child class method which is overriding is called the overriding method.
- In overriding method resolution is always takes care by JVM based on runtime object hence overriding is also considered as runtime polymorphism or dynamic polymorphism or late binding.
- The process of overriding method resolution is also known as dynamic method dispatch.
- In overriding method names and arguments must be the same. That is, method signatures must be the same.
- Until 1.4 version the return types must be same but from 1.5 version onwards covariant return types are allowed.
- According to this Child class method return type need not be same as Parent class method return type and Child types are also allowed.

Constructor Execution in case of inheritance

In case of Constructor the Parent class constructor would be executed followed by Child class constructor with the help of “super()”.
It is basically used to make a call to the parent class constructor.
Internally jvm uses super() to promote constructor chaining at inheritance level.

Example

```
class Parent{
             int x=10;
            {
                   methodOne();
                   System.out.println("Parent first instance block");
            }
            Parent(){
                    System.out.println("Parent class constructor");
            }

            public static void main(String... args){
                   Parent p=new Parent();
                   System.out.println("Parent class main()");
            }
            public void methodOne(){
                   System.out.println(y);
            }
            int y=20;
}
class Child extends Parent{
      int i=100;
      {
              methodTwo();
              System.out.println("Child first instance block");
      }
      Child(){
             System.out.println("Child class constructor");
      }

      public static void main(String... args){
             Child c=new Child();
             System.out.println("Child class main()");
      }
      public void methodTwo(){
              System.out.println(j);
      }
      int j=200;
}
public class Test {
       public static void main(String[] args) {

       }
}
```

Rule

Whenever we create child class Object,the following things take place

a. Identification of instance variables,instance blocks from parent to child.
b. Execution of instance variables assignment,instance block only of parent class.
c. Execution of parent class constructor.
d. Execution of instance variables assignment,instance block only of child class.
e. Execution of child class constructor.

Output

java Parent
0
Parent first instance block
Parent class constructor
Parent class main()

Output

java Child
0
Parent first instance block
Parent class constructor
0
Child first instance block
Child class constructor
Child class main()