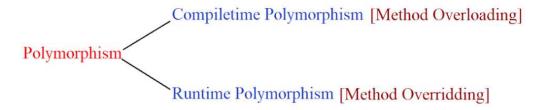
<u>Polymorphism:</u> It is a process of performing multiple tasks with the same identity.



Method Overloading: "Writing two or more methods in the same class having

same method name but different parameters."

Method Overridding: "Writing two or more methods in '2' different Classes having same

method name, same parameters & same returntype."

1) If we want to perform Method overriding 100%, we need to achieve Inheritance

- 2) If we can't inherit a method we can't override. (Ex : private methods)
- 3) private > default > protected > public
- 4) Whenever we are performing method overriding child class method should not be more restrictive than the parent class method
- 5) We cannot override static method, It may seem like we are overriding but that concept is known as method hiding
- 6) In method hiding child class method will hide the implementation of the parent class method
- 7) After jdk 1.5 return types may not be same in co-variant return types. (Co-variant return type concept is applicable only for object types but not for primitives)

Method Overriding

- "Writing two or more methods in super and sub classes with the same name and same signature is called Method Overriding".
- It is also known as 'late-binding' or 'run-time' polymorphism.
- The method present in super class is called overridden method and the method present in the sub class is called over ridding method.
- When an overridden method is called through a super class reference, Java
 determines which version of that method is to be executed based upon the
 type of the object being referred to at the time the call occurs. Thus, this
 determination is made at run time. That is the reason it is called Run-time
 polymorphism.

Overridden Rules

- ➤ The argument list must exactly match that of the overridden method.
- The return type should exactly match that of the overridden method (upto 1.4V).
- ➤ After jdk 1.5 return types may not be same in co-variant return types. (Co-variant return type concept is applicable only for object types but not for primitives)
- ➤ The access level must not be more restrictive that that of the over ridden method. (private < default < protected < public)
- If a method can't be inherited we cannot override it (Ex: private).

- We can't override a static method as non static methods and vice versa.
- ➤ If you are over ridding two static methods then it will be "method hiding".
- ➤ The overriding method must not throw new broader checked exceptions than those declared by the overridden method (to be discussed later).
- > For unchecked exceptions there are no restrictions.

Overloading Vs Overriding

Method Overloading	Method Overriding
It occurs with in the same class	It occurs with in the super class and sub class
Inheritance is not involved since it deals with only one class	Inheritance is involved because it occurs between Super and Sub classes
In Overloading Return type need not be the same	In Overriding Return type must be same
Parameters must be different when we do Overloading	Parameters must be same
In Overloading one method can't hide another method	In Overriding sub class method hides the super class methods

Private methods we cannot inherit, so we cannot override.

Overloading does not need inheritance.

Overloading occurs in the same class.

Overriding occurs in the 2 different classes.

```
1 package com.pack1;
                                                         1 package com.pack1;
                                                                                                                 <terminated> Cla
                                                                                                                 10
3 public class ClassA
                                                         3 public class ClassB extends ClassA
4 {
                                                         5=
 5=
       void meth1()
                                                                void meth1()
 6
                                                         6
                                                                {
 7
           System.out.println("Class A meth1()");
                                                                    System.out.println(10);
8
                                                         8
9 }
                                                         9=
                                                                public static void main(String[] args)
10 /*
                                                                    ClassA aobj=new ClassB();
11 1) If we want to perform Method overridding 100
                                                        811
12 2) If we cant inherit a method we cant override
                                                        812
                                                                    aobj.meth1();
                                                        13
14 */
                                                        14 }
15
                                                        15
```

Actually, in the above case we are calling method of class A but here Class B method is executed this is called method overriding.

```
cobj1 : com.pack1.ClassC@3feba861
 1 package com.pack1;
                                                                                      cobj1 : com.pack1.ClassC@3feba861
 3 public class ClassC
                                                                                      cobj2 : com.pack1.ClassC@b4c966a
      public static void main(String[] args)
           ClassC cobj1=new ClassC();
 8
          ClassC cobj2=new ClassC();
 9
10
           System.out.println("cobj1 : "+cobj1);
          System.out.println("cobj1 : "+cobj1.toString());
11
12
           System.out.println("\ncobj2 : "+cobj2);
13
           System.out.println("cobj2 : "+cobj2.toString());
                                                                       Ι
      }
15
16 }
17 /*
       Whenever we are printing an Object of a class,
19
       internally compiler is going to call toString()
20
       which is present in the Object class
21 */
```

```
cobj1 : Java is awesome!!!
 1 package com.pack1;
 3 public class ClassC
                                                                                         obj2 : Java is awesome!!!
 4 {
       @Override // Annotation
                                                                                        cobj2 : Java is aweso<mark>me!!!</mark>
5-
       public String toString()
 7
8
           return "Java is awesome!!!";
9
10-
      public static void main(String[] args)
11
12
           ClassC cobj1=new ClassC();
13
          ClassC cobj2=new ClassC();
14
          System.out.println("cobj1 : "+cobj1);
           System.out.println("cobj1 : "+cobj1.toString());
16
17
           System.out.println("\ncobj2 : "+cobj2);
18
           System.out.println("cobj2 : "+cobj2.toString());
19
20
21 }
22 /*
23
        Whenever we are printing an Object of a class,
        internally compiler is going to call toString()
```

In the above case we are inheriting object class internally.

An object class is parent class for all the classes in java by default we need not inherit(extends) or import

To override to string() in child class we are writing same method name by this we are overriding to String() from object class.

Why do we need to override?

I am not satisfied with that method implementation I want to write my own implementation.

Then why can't you take the new method name.

There are some situations where you cannot change the method name, but we can change its method implementation.

4key points in inheritance

1 is for has a relationship

2 is for method overriding

3 is for is a relationship (actual inheritance)

4th key point invalid

Method overriding is also known as late-binding or run time polymorphism.

When an overridden method is called through parent class reference java determines which version of that method is to be called based upon the type of object being referred to whenever you're calling that method thus this determination will done at runtime.

```
# D + P.
  1 package com.pack1;
                                                              1 package com.pack1;
                                                                                                                                      <terminated> ClassB [Jav
                                                                                                                                      Class A meth1()
   public class ClassA
                                                                public class ClassB extends ClassA
                                                            5 6
        void meth1() // overridden method
                                                                     void meth1() // overridding method
            System.out.println("Class A meth1()");
                                                                         System.out.println(10);
 9-
        void meth2(int x)
                                                              Q=
                                                                     @Override
10
                                                            -10
                                                                     void meth2(int num)
11
            System.out.println("Class A meth2()");
                                                             11
                                                                    {
12
                                                             12
                                                                         System.out.println(20);
13 }
                                                             13
14 /*
                                                             14
                                                                    public static void main(String[] args)
15
    1) If we want to perform Method overridding 100%
                                                             15
16 2) If we cant inherit a method we cant override.
                                                             16
                                                                         ClassA aobj1=new ClassA(); // 1st-point
17
                                                             17
                                                                         aobj1.meth1();
18 */
                                                             18
                                                                                                                                              I
19
                                                             19
                                                                         ClassA aobj2=new ClassB(); // 2nd-point
20
                                                             20
                                                                         aobj2.meth1();
21
                                                             21
                                                                         aobj2.meth2(100);
22
                                                             22
23
                                                             23 }
 1 package com.pack1;
                                                              void meth1() // overridding method
                                                                                                                          <terminated> ClassB [Java
  public class ClassA
                                                                                                                          Class A meth1()
                                                                  System.out.println(10);
       void meth1() // overridden method
                                                              @Override
                                                              void meth2(int num)
                                                       -10
          System.out.println("Class A meth1()");
                                                        12
                                                                  System.out.println(20);
       void meth2(int x)
                                                        13
           System.out.println("Class A meth2()");
11
                                                               void meth3(int i, String msg)
12
                                                       16
      void meth3(int x, String s)
13
                                                                  System.out.println(30);
          System.out.println("Class A meth3()");
15
                                                        19
16
                                                       20
                                                              public static void main(String[] args)
17 }
   1) If we want to perform Method overridding 100%
19
   2) If we cant inherit a method we cant override.
                                                                  ClassA aobj2=new ClassB(); // 2nd-point
22
                                                                  aobj2.meth1();
23
                                                                  aobj2.meth2(100);
24
                                                                   aobj2.meth3(100, "Java");
                                                        30 }
```

Is it mandatory that both access modifiers of method should be same?

The answer is no

Can we override static method with a non-static method?

The answer is no

Can we override static method with a static method?

The answer is no

```
protected void meth2(int x)
                                                                                                                                     <terminated> ClassB [Java .
10
                                                                           System.out.println(20):
                                                                                                                                     Class A meth1()
            System.out.println("Class A meth2()");
12
       void meth3(int x, String s)
13-
                                                                       void meth3(int i, String msg)
                                                                                                                                     Class A meth3()
15
            System.out.println("Class A meth3()");
                                                                           System.out.println(30);
       static void meth4()
                                                                       static void meth4()
            System.out.println("Class A meth3()");
                                                                           System.out.println(40);
21 }
                                                                       public static void main(String[] args)
23
   1) If we want to perform Method overridding 100% we
   2) If we cant inherit a method we cant override. (E
                                                                           ClassA aobj1=new ClassA(); // 1st-point

3) private > default > protected > public
4) Whenever we are performing method overriding chi
                                                                           aobj1.meth1();
       should not be more restrictive than the parent c
                                                                           ClassA aobj2=new ClassB(); // 2nd-point
28
29
                                                                           aobj2.meth1();
                                                                           aobj2.meth2(100);
                                                                           aobj2.meth3(100, "Java");
30
31
                                                                           aobj2.meth4();
32
33
```

Static belongs to the class not to the object

It may seem like you are overriding but no, this concept is known as a method hiding. We should not write annotation because we will get an error.

In method hiding child class method will hide the implementation of the parent class method.

```
public void meth2(int num)
                                                                                                                                          A - - - -
        protected void meth2(int x)
                                                                           11
                                                                                                                                             <terminated> ClassB [Java
                                                                                        System.out.println(20);
                                                                                                                                             Class A meth1()
11
             System.out.println("Class A meth2()");
12
                                                                           140
        void meth3(int x, String s)
                                                                                   void meth3(int i, String msg)
                                                                           -15
                                                                                                                                             Class A meth4()
             System.out.println("Class A meth3()");
15
                                                                                        System.out.println(30);
16
                                                                           18
        static void meth4()
                                                                           19-
                                                                                   static void meth4()
18
                                                                           20
             System.out.println("Class A meth4()");
19
                                                                           21
22
23°
24
25
26
27
                                                                                        System.out.println(40);
 20
        ClassA meth5()
22
23
24
             System.out.println("Class A meth5()");
                                                                                        System.out.println(50);
             return new ClassA();
                                                                                        return new ClassA():
26 }
                                                                                   public static void main(String[] args)
                                                                           29
    1) If we want to perform Method overridding 100% we need
                                                                           30
                                                                                        ClassA aobj1=new ClassA(); // 1st-point
                                                                                        aobj1.meth1();
    2) If we cant inherit a method we cant override. (Ex : pri

    private > default > protected > public
    Whenever we are performing method overriding child class

                                                                           32
                                                                                        ClassA aobj2=new ClassB(); // 2nd-point
                                                                           33
        should not be more restrictive than the parent class m\varepsilon
                                                                                        aobj2.meth1();
    5) We cannot override static method,It may seem like we ar
6) In method hiding child class method will hide the imple
*/
                                                                                        aobj2.meth2(100);
                                                                           36
                                                                                        aobj2.meth3(100,
                                                                          237
                                                                                        aobj2.meth4();
                                                                                        aobj2.meth5();
> ≨ Iraining > ≛ src > ∄ com.pack1 + ⊖ ClassA > ▲ meth5() : ClassA
                                                                           raining * src * # com.pack1 * # Class8 * * meth4() : void public void metn2(int num)
                                                                                                                                          protected void meth2(int x)
                                                                           11
                                                                                                                                             <terminated> ClassB [Java
                                                                           12
                                                                                        System.out.println(20);
                                                                                                                                             Class A meth1()
 11
             System.out.println("Class A meth2()");
 12
                                                                           14=
 13-
        void meth3(int x, String s)
                                                                                    void meth3(int i, String msg)
 14
                                                                                                                                             Class A meth4()
 15
             System.out.println("Class A meth3()");
                                                                           17
                                                                                        System.out.println(30);
 16
                                                                           18
                                                                          19=
20
21
22
        static void meth4()
                                                                                    static void meth4()
 18
 19
             System.out.println("Class A meth4()");
                                                                                        System.out.println(40);
 20
        ClassA meth5() -
                                                                                   ClassB meth5() -
822
                                                                           24
             System.out.println("Class A meth5()");
                                                                           25
                                                                                        System.out.println(50);
                                                                           26
             return new ClassA();
                                                                                        return new ClassB();
25
 26 }
                                                                           28-
                                                                                   public static void main(String[] args)
 27 /*
                                                                           29
    1) If we want to perform Method overridding 100% we need
                                                                           30
                                                                                        ClassA aobj1=new ClassA(); // 1st-point
    2) If we cant inherit a method we cant override. (Ex : pri
3) private > default > protected > public
                                                                                        aobj1.meth1();
                                                                           32
    4) Whenever we are performing method overriding child class
                                                                                       ClassA aobi2=new ClassB(): // 2nd-point
        should not be more restrictive than the parent class m\varepsilon
                                                                                        aobj2.meth1();
    5) We cannot override static method, It may seem like we ar
    6) In method hiding child class method will hide the imple ^*/
 33
                                                                           35
                                                                                        aobj2.meth2(100);
                                                                                        aobj2.meth3(100, "Java");
                                                                           36
                                                                                        aobj2.meth4();
```

Considering the above two images of method 5

This process is known as covariant return types applicable for classes not for primitive data types

After jdk 1.5 return types may not be same in co-variant return types. (Co-variant return type concept is applicable only for object types but not for primitives)

Can we override a constructor?

No, we have 2 reasons

- 1. We cannot inherit a constructor, without inheritance we cannot override.
- 2. In class A the constructor name will be class A and in class B the constructor name will be Class B, these are not the same.

```
1 package com.pack1;
                                                                3 public class ClassB extends ClassA
                                                                4 {
  public class ClassA
                                                                     protected void meth1() // overridding method
 5=
      void meth1() // overridden method
                                                                         System.out.println(10);
 6
          System.out.println("Class A meth1()");
8
                                                                     public void meth2(int num)
      protected void meth2(int x)
                                                               11
10
                                                                         System.out.println(20);
          System.out.println("Class A meth2()");
11
12
      void meth3(int x, String s)
13=
                                                              -15
                                                                     void meth3(int i, String msg)
14
                                                               16
          System.out.println("Class A meth3()");
15
                                                                         System.out.println(30);
                                                               17
16
                                                               18
17
      static void meth4()
                                                               199
                                                                     static void meth4()
18
                                                               20
          System.out.println("Class A meth4()");
19
                                                               21
                                                                         System.out.println(40);
20
                                                               22
21=
      ClassA meth5()
                                                              -23⊕
                                                                     ClassB meth5()
22
23
          System.out.println("Class A meth5()");
                                                               25
                                                                         System.out.println(50);
                                                               26
24
          return new ClassA();
                                                                         return new ClassB();
25
                                                               27
26 }
                                                                     public static void main(String[] args)
29
               ClassA aobj1=new ClassA(); // 1st-point
30
                aobj1.meth1();
31
               ClassA aobj2=new ClassB(); // 2nd-point
33
34
                aobj2.meth1();
                aobj2.meth2(100);
                aobj2.meth3(100, "Java");
36
-37
                aobj2.meth4();
38
                aobj2.meth5();
39
```

```
1 package com.pack1;
3 public class ClassC
4 {
5=
      @Override // Annotation
6
      public String toString()
 7
           return "Java is awesome!!!";
8
9
      public static void main(String[] args)
100
11
12
          ClassC cobj1=new ClassC();
13
          ClassC cobj2=new ClassC();
14
          System.out.println("cobj1 : "+cobj1);
15
          //System.out.println("cobj1 : "+cobj1.toString());
17
          System.out.println("\ncobj2: "+cobj2);
18
19
          //System.out.println("cobj2 : "+cobj2.toString());
20
21 }
22 /*
        Whenever we are printing an Object of a class,
23
24
        internally compiler is going to call toString()
25
       which is present in the Object class
26 +/
```

private > default > protected > public

parent class

private invalid(inheritance)

public-----public

protected -----protected & public

default ----- default, protected & public

```
<terminated> ClassC [Java Application] C:\l
 1 package com.pack1;
                                                          cobj1 : Java is awesome!!!
 3 public class ClassC
                                                          cobj2 : Java is awesome!!!
 4 {
       @Override // Annotation
 5=
6
       public String toString()
 7
           return "Java is awesome!!!";
 8
 9
       public static void main(String[] args)
10
11
12
       ___ Object cobj1=new ClassC();
13
       Dbject cobj2=new ClassC();
14
15
           System.out.println("cobj1 : "+cobj1);
           //System.out.println("cobj1 : "+cobj1.toSt
16
17
18
           System.out.println("\ncobj2 : "+cobj2);
19
           //System.out.println("cobj2 : "+cobj2.toSt
       }
20
21 }
22 /*
        Whenever we are printing an Object of a class
23
        internally compiler is going to call toString
24
25
        which is present in the Object class
26 */
```

Can the final method override?

The Answer is no (wait for the coming classes)