SYNTAX TECHNOLOGIES

JAVA

Class 24

Agenda

Method Overriding

Rules for Method Overriding

Task

1. Create a class named 'Programming'. While creating an object of the class, if nothing is passed to it, then the message "I love programming languages" should be printed. If some String is passed to it, then in place of "programming languages" the value variable should be printed. Example, if we pass "Java", then "I love Java" should be printed.
2. Create 1 class with a static method that has 3 overloaded forms. Then call each overloaded method with specific arguments and observe result.
3. Create 1 class with a private method that has 3 overloaded forms. Then call each overloaded method with specific arguments and observe result.

Runtime / Dynamic

Runtime polymorphism can be achieved by using Method overriding

Method of superclass is overridden in subclass to provide more specific implementation.

Dynamic binding always says create an object of base class but do not create the object of derived classes.

The process of binding appropriate versions (overridden method) of derived classes which are inherited from base class with base class object is known as dynamic binding.

Runtime / Dynamic

Advantages of dynamic binding along with polymorphism with method overriding:

- Less memory space
- Less execution time
- More performance

Method Overriding in Java

Whenever same method name is existing in both base class and derived class with same types of parameters or same order of parameters is known as **method Overriding**.

If subclass (child class) has the same method as declared in the parent class, it is known as **method overriding in java**.

Note: Without Inheritance method overriding is not possible.

Method Overriding in Java

Advantage of Java Method Overriding

Method Overriding is used to provide specific implementation of a method that is already provided by its super class.

Method Overriding is used for Runtime Polymorphism

Note: Without Inheritance method overriding is not possible.

Method Overriding in Java

Rules for Method Overriding

- method must have same name as in the parent class and child class.
- method must have same parameter as in the parent class.
- must be IS-A relationship (inheritance).

Note: Without Inheritance method overriding is not possible.

Example of method overriding in Java

We have defined the walk method in the subclass as defined in the parent class but it has some specific implementation.
The name and parameter of the method is same and there is IS-A relationship between the classes, so there is method overriding.

```
class Walking {
    void walk() {
    SOP("Man walking fastly");
    }
}
class Man extends walking {
     void walk() {
     SOP("Man walking slowly");
     }
}
```

```
class OverridingDemo {
     public static void main(String args[]) {
     Man obj = new Man();
     obj.walk();
     }
}
```

Note: Whenever we are calling overridden method using derived class object reference the highest priority is given to current class (derived class). We can see in the above example high priority is derived class.

Rules for Java Method Overriding

- If a method cannot be inherited then it cannot be overridden. Therefor, private methods cannot be overridden.

- The method signature i.e. method name, parameter list and return type have to match exactly in super and subclass.

- The access level **cannot be more restrictive** than the overridden method access level. For example: if the super class method is declared public then the overriding method in the sub class cannot be either private or public.

Task

1. Create a class 'Degree' having a method 'getDegree' that prints "I got a degree". Class 'Degree' has two subclasses namely 'Undergraduate' and 'Postgraduate' each having a method with the same name that prints "I am an Undergraduate" and "I am a Postgraduate" respectively. Call the method by creating an object of each of the three classes.