Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Lab Number:	8
Student Name:	Siddhant Kedar
Roll No :	E-05

Title: Method Overriding in java Inheritance

Learning Objective:

- Students will be able to execute a simple method Overriding programs in Java.

Learning Outcome:

Understanding method overriding in Java

Theory:

Method Overriding in java : Method overriding is a process of overriding base class method by derived class method with more specific definition.

Method overriding performs only if two classes have is-a relationship. It mean class must have inheritance. In other words, It is performed between two classes using inheritance relation.

In overriding, method of both class must have same name and equal number of parameters.

Method overriding is also referred to as runtime polymorphism because calling method is decided by JVM during runtime.

The key benefit of overriding is the ability to define method that's specific to a particular subclass type.

Rules of Method Overriding in Java:

1. Method name must be same for both parent and child classes.

2. Access modifier of child method must not restrictive than parent class method.

3. Private, final and static methods cannot be overridden.

4. There must be an IS-A relationship between classes (inheritance).

Faculty: Ms. Deepali Kayande

Program 1:

Algorithm:

1) Start

2) Create class Animal

3) Method in superclass

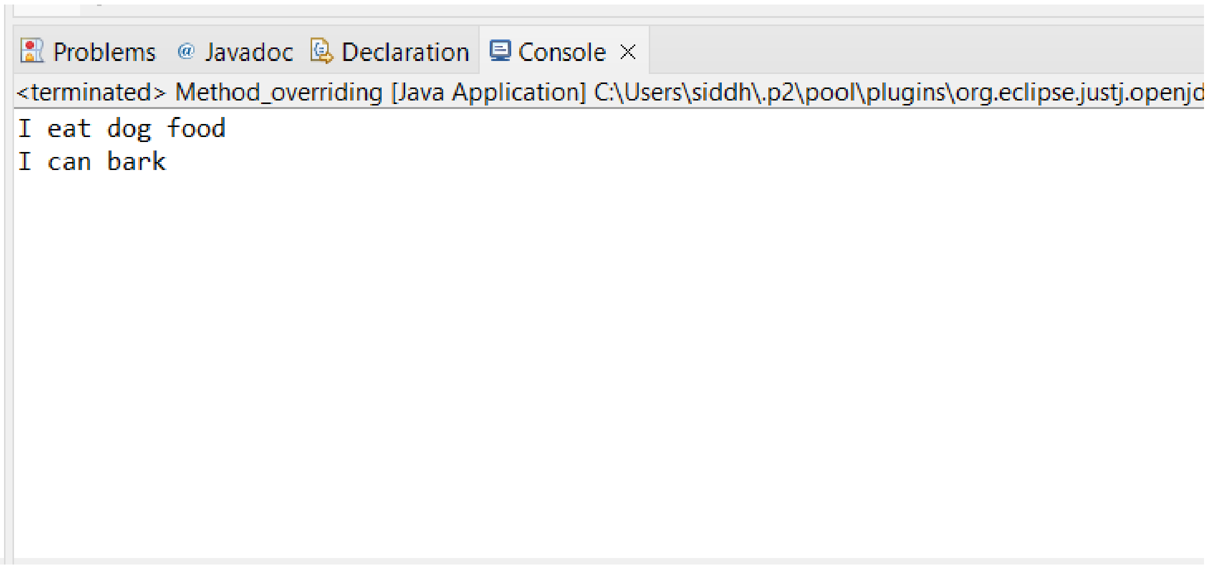
4) Overriding the eat() method

5) Create a new method in subclass and an object of subclass

6) Call the eat() method

```
class Animal {

  // method in the superclass
  public void eat() {
    System.out.println("I can eat");
  }
}

// Dog inherits Animal
class Dog extends Animal {

  // overriding the eat() method
  @Override
  public void eat() {
    System.out.println("I eat dog food");
  }

  // new method in subclass
  public void bark() {
    System.out.println("I can bark");
  }
}


class  Method_overriding {
  public static void main(String[] args) {


    // create an object of the subclass
    Dog labrador = new Dog();


    // call the eat() method
    labrador.eat();
    labrador.bark();
  }
}
```

Output:

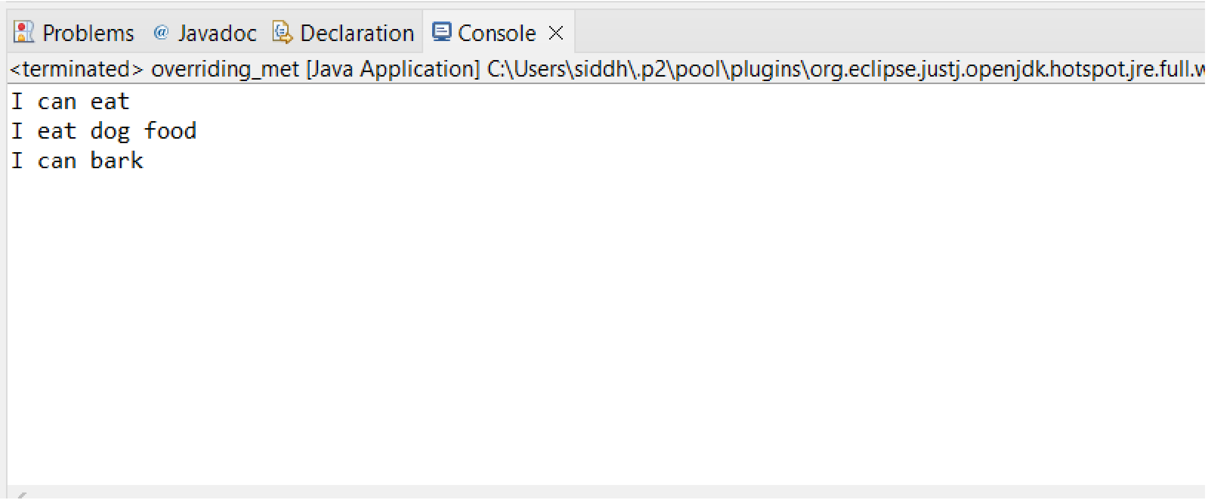


```
<terminated> Method_overriding [Java Application] C:\Users\siddh\.p2\pool\plugins\org.eclipse.justj.openjd
I eat dog food
I can bark
```

Program 2:

```
class Animal {
  // method in the superclass
  public void eat() {
    System.out.println("I can eat");
  }
}

// Dog inherits Animal
class Dog extends Animal {

  // overriding the eat() method
  @Override
  public void eat() {

    // call method of superclass
    super.eat();
    System.out.println("I eat dog food");
  }

  // new method in subclass
  public void bark() {
    System.out.println("I can bark");
  }
}

public class overriding_met {
  public static void main(String[] args) {
    // create an object of the subclass
    Dog labrador = new Dog();


    // call the eat() method
    labrador.eat();
    labrador.bark();
  }
}
```

Output:



```
<terminated> overriding_met [Java Application] C:\Users\siddh\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.w
I can eat
I eat dog food
I can bark
```