JAVA

Polymorphism

01

POLYMORPHISM

Polymorphism is the ability of a single object to take on multiple forms. In Java, polymorphism is implemented using inheritance, method overloading, and method overriding.

METHOD OVERLOADING

Compile time polymorphism

Method overloading is the ability to create multiple methods with the same name but **different parameters**. For example:

```java
public class Calculator {
  public int add(int x, int y) {
    return x + y;
  }

  public int add(int x, int y, int z)
{   return x + y + z;
  }
}
```

In this example, the add method is overloaded with two different versions: one that takes two arguments and one that takes three arguments.

METHOD OVERRIDING

Run time polymorphism

Method overriding is the ability to create a method in a subclass with the **same name** and **parameters** as a method in the superclass, and then provide a **different implementation** for the method in the subclass. For example:

```java
public class Animal {
  public void makeSound() {
    System.out.println("Some generic animal sound");
  }
}

public class Dog extends Animal {
  @Override
  public void makeSound() {
    System.out.println("Woof!");
  }
}
```

In this example, the makeSound method is overridden in the Dog class to provide a specific implementation for dogs. When the makeSound method is called on a Dog object, the version defined in the Dog class will be used, rather than the version defined in the Animal class.

Gunjan Jain