

Solve Coding Problems

Java Tutorial

Overview of Java

Basics of Java

Input/Output in Java

Flow Control in Java

Operators in Java

Strings in Java

Arrays in Java

OOPS in Java

Inheritance in Java

Abstraction in Java

Encapsulation in Java

Polymorphism in Java

Constructors in Java

Methods in Java

Static methods vs Instance methods in Java

Abstract Method in Java with Examples

Overriding in Java

Method Overloading in Java

Difference Between Method Overloading and Method Overriding in Java

Interfaces in Java

Wrapper Classes in Java

Keywords in Java

Access Modifiers in Java

Memory Allocation in Java

Classes of Java

Packages in Java

Java Collection Tutorial

Exception Handling in Java

Multithreading in Java

Synchronization in Java

File Handling in Java

Java Regex

Java IO

Java Networking

JDBC - Java Database Connectivity

Java 8 Features - Complete Tutorial

## Difference Between Method Overloading and Method Overriding in Java

The differences between Method Overloading and Method Overriding in Java are as follows:

Method Overloading	Method Overriding
Method overloading is a compile-time polymorphism.	Method overriding is a run-time polymorphism.
Method overloading helps to increase the readability of the program.	Method overriding is used to grant the specific implementation of the method which is already provided by its parent class or superclass.
It occurs within the class.	It is performed in two classes with inheritance relationships.
Method overloading may or may not require inheritance.	Method overriding always needs inheritance.
In method overloading, methods must have the same name and different signatures.	In method overriding, methods must have the same name and same signature.
In method overloading, the return type can or can not be the same, but we just have to change the parameter.	In method overriding, the return type must be the same or co-variant.
Static binding is being used for overloaded methods.	Dynamic binding is being used for overriding methods.
Poor Performance due to compile time polymorphism.	It gives better performance. The reason behind this is that the binding of overridden methods is being done at runtime.
Private and final methods can be overloaded.	Private and final methods can't be overridden.
The argument list should be different while doing method overloading.	The argument list should be the same in method overriding.

### Method Overloading in Java

[Method Overloading](#) is a Compile time polymorphism. In method overloading, more than one method shares the same method name with a different signature in the class. In method overloading, the return type can or can not be the same, but we have to change the parameter because, in java, we can not achieve method overloading by changing only the return type of the method.

Example of Method Overloading:

Java

```
// Java Program to Implement
// Method Overloading
import java.io.*;

class MethodOverloadingEx {

    static int add(int a, int b) { return a + b; }

    static int add(int a, int b, int c)
    {
        return a + b + c;
    }

    // Main Function
    public static void main(String args[])
    {
        System.out.println("add() with 2 parameters");
        // Calling function with 2 parameters
        System.out.println(add(4, 6));

        System.out.println("add() with 3 parameters");
        // Calling function with 3 Parameters
        System.out.println(add(4, 6, 7));
    }
}
```

Output

```
add() with 2 parameters
10
add() with 3 parameters
17
```

### Method Overriding in Java

[Method Overriding](#) is a Run time polymorphism. In method overriding, the derived class provides the specific implementation of the method that is already provided by the base class or parent class. In method overriding, the return type must be the same or co-variant (return type may vary in the same direction as the derived class).

Example of Method Overriding:

Java

```
// Java Program to implement
// Method Overriding
import java.io.*;

// Base Class
class Animal {
```

[Skip to content](#)

```
void eat()
{
    System.out.println("eat() method of base class");
    System.out.println("eating.");
}

// Inherited Class
class Dog extends Animal {
    void eat()
    {
        System.out.println("eat() method of derived class");
        System.out.println("Dog is eating.");
    }
}

class MethodOverridingEx {
    // Main Function
    public static void main(String args[])
    {
        Dog d1 = new Dog();
        Animal a1 = new Animal();

        d1.eat();
        a1.eat();

        Animal animal = new Dog();
        // eat() method of animal class is overridden by
        // base class eat()
        animal.eat();
    }
}
```

Output

```
eat() method of derived class
Dog is eating.
eat() method of base class
eating.
eat() method of derived class
Dog is eating.
```

Output explanation: Here, we can see that a method eat() has overridden in the derived class name Dog that is already provided by the base class name Animal. When we create the instance of class Dog and call the eat() method, we see that only derived class eat() method run instead of base class method eat(), and When we create the instance of class Animal and call the eat() method, we see that only base class eat() method run instead of derived class method eat().

Feeling lost in the vast world of Backend Development? It's time for a change! Join our [Java Backend Development - Live Course](#) and embark on an exciting journey to master backend development efficiently and on schedule.

What We Offer:

- Comprehensive Course
- Expert Guidance for Efficient Learning
- Hands-on Experience with Real-world Projects
- Proven Track Record with 100,000+ Successful Geeks

Last Updated : 25 May, 2023

146

[Previous](#)

[Next](#)

[Method Overloading in Java](#)

[Differences between Interface and Class in Java](#)

Share your thoughts in the comments

[Add Your Comment](#)

### Similar Reads

[Difference between Method Overloading and Method Overriding in Python](#)

[Function Overloading vs Function Overriding in C++](#)

[Exception Handling with Method Overriding in Java](#)

[Different Ways to Prevent Method Overriding in Java](#)

[Java Program to Use Method Overriding in Inheritance for Subclasses](#)

[Java - Covariant Method Overriding with Examples](#)

[When We Need to Prevent Method Overriding in Java ?](#)

[Overriding equals method in Java](#)

[Overriding toString\(\) Method in Java](#)

[Variables in Java Do Not Follow Polymorphism and Overriding](#)

M MKS075

Article Tags : [Java-Overloading](#), [java-overriding](#), [Difference Between](#), [Java](#)

Practice Tags : [Java](#)



Legal  
Careers  
In Media  
Contact Us  
Advertise with us  
GFG Corporate Solution  
Placement Training Program

GGG Weekly Contest  
DSA in JAVA/C++  
Master System Design  
Master CP  
GeeksforGeeks Videos  
Geeks Community

Java  
C++  
PHP  
GoLang  
SQL  
R Language  
Android Tutorial  
Tutorials Archive

Algorithms  
DSA for Beginners  
Basic DSA Problems  
DSA Roadmap  
Top 100 DSA Interview Problems  
DSA Roadmap by Sandeep Jain  
All Cheat Sheets

Data Science For Beginner  
Machine Learning Tutorial  
ML Maths  
Data Visualisation Tutorial  
Pandas Tutorial  
NumPy Tutorial  
NLP Tutorial  
Deep Learning Tutorial

CSS  
Web Templates  
CSS Frameworks  
Bootstrap  
Tailwind CSS  
SASS  
LESS  
Web Design  
Django Tutorial

Python Tutorial

Python Programming Examples  
Python Projects  
Python Tkinter  
Web Scraping  
OpenCV Tutorial  
Python Interview Question

Computer Science

Operating Systems  
Computer Network  
Database Management System  
Software Engineering  
Digital Logic Design  
Engineering Maths

DevOps

Git  
AWS  
Docker  
Kubernetes  
Azure  
GCP  
DevOps Roadmap

Competitive Programming

Top DS or Algo for CP  
Top 50 Tree  
Top 50 Graph  
Top 50 Array  
Top 50 String  
Top 50 DP  
Top 15 Websites for CP

System Design

High Level Design  
Low Level Design  
UML Diagrams  
Interview Guide  
Design Patterns  
OOD  
System Design Bootcamp  
Interview Questions

JavaScript

JavaScript Examples  
TypeScript  
ReactJS  
NextJS  
AngularJS  
NodeJS  
Lodash  
Web Browser

Preparation Corner

Company-Wise Recruitment Process  
Resume Templates  
Aptitude Preparation  
Puzzles  
Company-Wise Preparation

School Subjects

Mathematics  
Physics  
Chemistry  
Biology  
Social Science  
English Grammar  
World GK

Management & Finance

Management  
HR Management  
Finance  
Income Tax  
Organisational Behaviour  
Marketing

Free Online Tools

Typing Test  
Image Editor  
Code Formatters  
Code Converters  
Currency Converter  
Random Number Generator  
Random Password Generator

More Tutorials

Software Development  
Software Testing  
Product Management  
SAP  
SEO - Search Engine Optimization  
Linux  
Excel

GeeksforGeeks Videos

DSA  
Python  
Java  
C++  
Data Science  
CS Subjects