**Sir git link**

[**https://github.com/checkcc/corejava.git**](https://github.com/checkcc/corejava.git)

**Java Features**

[**https://www.oracle.com/java/technologies/javase/features.html**](https://www.oracle.com/java/technologies/javase/features.html)

**Java**

[**https://www.slideshare.net/SpotleAI/java-data-types-191743412**](https://www.slideshare.net/SpotleAI/java-data-types-191743412)

[**https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html**](https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html)

**Interface:**

[**https://www.javatpoint.com/interface-in-java**](https://www.javatpoint.com/interface-in-java)

[**https://www.tutorialspoint.com/java/java\_interfaces.htm**](https://www.tutorialspoint.com/java/java_interfaces.htm)

**Multiple Inheritance region**

[**https://www.tutorialspoint.com/what-is-diamond-problem-in-case-of-multiple-inheritance-in-java**](https://www.tutorialspoint.com/what-is-diamond-problem-in-case-of-multiple-inheritance-in-java)

**Type casting**

[**https://www.slideshare.net/farooqkhan125/type-casting-in-java-83416084**](https://www.slideshare.net/farooqkhan125/type-casting-in-java-83416084)

[**https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html**](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html)

**List of exception**[**https://programming.guide/java/list-of-java-exceptions.html**](https://programming.guide/java/list-of-java-exceptions.html)

**Data Types**

<https://www.javatpoint.com/java-data-types>

<https://www.javatpoint.com/java-data-types>

<https://www.slideshare.net/Thenmurugeshwari/operators-in-java>

**Is and Has Relation**

[What is Has-A-Relation in Java? - GeeksforGeeks](https://www.geeksforgeeks.org/what-is-has-a-relation-in-java/)

**String :**

[**https://docs.oracle.com/javase/8/docs/api/java/lang/String.html**](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html)

[**https://www.javatpoint.com/difference-between-stringbuffer-and-stringbuilder**](https://www.javatpoint.com/difference-between-stringbuffer-and-stringbuilder)

[**https://www.javatpoint.com/string-pool-in-java**](https://www.javatpoint.com/string-pool-in-java)

[**https://www.journaldev.com/797/what-is-java-string-pool**](https://www.journaldev.com/797/what-is-java-string-pool)[**https://www.javatpoint.com/understanding-toString()-method**](https://www.javatpoint.com/understanding-toString()-method)

[**https://docs.oracle.com/javase/7/docs/api/java/util/StringTokenizer.html**](https://docs.oracle.com/javase/7/docs/api/java/util/StringTokenizer.html)

**Thread:**

[**https://www.slideshare.net/PrabhakaranVM1/java-threads-123152980**](https://www.slideshare.net/PrabhakaranVM1/java-threads-123152980)

[**https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.html**](https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.html)

[**https://www.tutorialspoint.com/java/java\_multithreading.htm**](https://www.tutorialspoint.com/java/java_multithreading.htm)

[**https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.html#run--**](https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.html#run--)

**Data Structure**

[**https://www.javatpoint.com/data-structure-tutorial**](https://www.javatpoint.com/data-structure-tutorial)

[**https://docs.oracle.com/javase/8/docs/api/java/util/Collection.html**](https://docs.oracle.com/javase/8/docs/api/java/util/Collection.html)

**Collection**

[**https://www.slideshare.net/Hitesh-Java/collections-lists-sets-110298426**](https://www.slideshare.net/Hitesh-Java/collections-lists-sets-110298426)

[**https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html**](https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html)

[**https://www.geeksforgeeks.org/set-in-java/**](https://www.geeksforgeeks.org/set-in-java/)

**Map**

GET /v1/path/to/resource   
Host: www.example.gov.au  
Accept: application/json, text/javascript

Create new magazine within the /magazines resource

POST /magazines   
Host: example.gov.au  
Accept: application/json, text/javascript

{...}

PUT /magazines/1234  
Host: example.gov.au  
Accept: application/json, text/javascript

[ ... ]

DELETE /magazines/1234  
Delete the magazine resource /magazines/1234

DELETE /magazines/1234 HTTP/1.1  
Host: example.gov.au  
Accept: application/json, text/javascript

[**https://docs.oracle.com/javase/8/docs/api/java/util/Map.html**](https://docs.oracle.com/javase/8/docs/api/java/util/Map.html)

ConcurrentHashMap,  HashMap, Hashtable, IdentityHashMap, LinkedHashMap, TreeMap, WeakHashMap

**Java Queue**

[**https://www.journaldev.com/13244/java-queue**](https://www.journaldev.com/13244/java-queue)

<https://docs.oracle.com/javase/8/docs/api/java/io/package-summary.html>

<https://docs.oracle.com/javase/8/docs/api/java/io/InputStream.html>

**Final keyword**

**https://beginnersbook.com/2014/07/final-keyword-java-final-variable-method-class/**

**Functional interface**

**Lambda**

[**https://www.javatpoint.com/java-lambda-expressions**](https://www.javatpoint.com/java-lambda-expressions)

**Functional Interface**

Method reference is used to refer method of functional **interface**. It is compact and easy form of lambda expression. Each time when you are using lambda expression to just referring a method, you can replace your lambda expression with method reference. In **this** tutorial,

There are following types of method references in java:

Reference to a static method.  
Reference to an instance method.  
Reference to a constructor.

**Bi function**

[**https://docs.oracle.com/javase/8/docs/api/java/util/function/BiFunction.html**](https://docs.oracle.com/javase/8/docs/api/java/util/function/BiFunction.html)

**Streams**

<https://docs.oracle.com/javase/8/docs/api/java/util/stream/package-summary.html>

<https://docs.oracle.com/javase/8/docs/api/java/util/stream/Stream.html>

<https://www.javatpoint.com/java-8-stream>

<https://www.javatpoint.com/java-8-stream>

Predicate

<https://www.geeksforgeeks.org/java-8-predicate-with-examples/>

<https://www.javatpoint.com/difference-between-comparable-and-comparator>

* In java we have .jar files and .war files.
* Java 8 uses parallel processing.

**Static:** It gives single instance.

**Abstract :** It provide partial information.

**Constructor** : Constructor invoke from super class.

Ctrl+shif+F

Questinon

1. Why everything in java is string

git –version

git config --global user.email “[ak88a@gmail.com](mailto:ak88a@gmail.com)”

git config –global user.name “Ankit”

git config –list

git config --help