Core Java Tutorial – Learn Core Java By Example

Last Updated On: September 13, 2019 By Softwaretestingo Admin

Core Java Tutorial: After so many requests from the SoftwareTestingo visitors for the Core Java Tutorials, we have the plan to build a platform where everyone can **learn java online**. But we are building this platform for testers, so for that, we have shared the java tutorial which is required by a Java Selenium Testers. We have listed all of those topics which are required for java selenium testers. So finally we have prepared the complete list, as well as the tutorial post so that any beginner can follow those posts and **learn java online.** By following those posts anyone can learn java programming and will help you to learn the core java in no time.

This **core java tutorial** post contains all the links of all the covered topics in a systematic order starting from the java tutorials for beginners level to the advanced topics. We have prepared the tutorial in such a way that it will help all category peoples like a

college student who is interested to learn the Java programming online or a working employee who is interested to learn and implement those for building an application or in his/her work as well, on all these things these tutorials will help you.

Check Also: Java Interview Programs

Learn Core Java Tutorial

To Learn the Java programming which is one of the most demanded languages in the world, refer these tutorials in a given order. We keep in mind that these tutorials will helpful to all level of learners, so if you don't have any prior knowledge also you won't face any difficulty in understanding these tutorials.

Core Java Tutorial For Beginners

- History Of Java
- Difference Between JDK, JVM, and JVM
- Java Programming Languages Features
- Java Installation Windows
- Set Java_Home Variable
- Java Hello World Program
- Variables in Java
- Java Data Types
- Java Method
- Java Operators
- If-else in Java
- Switch Cases in Java
- Java for loop
- Java while loop
- Java do-while loop
- Java static keyword
- Java break keyword
- Java continue keyword

Core Java Tutorial – Object-Oriented Concepts

- Constructor in Java
- OOPS Concepts In Java
- Java Access Modifiers – public, protected, private and default
- **Composition in Java**
- Inheritance in Java & Types of inheritance in Java
- **Aggregation in Java**
- **Association in Java**
- Super Keyword in Java
- Method overloading in Java
- Method overriding in Java
- Polymorphism in Java & Types of polymorphism in Java
- Static and dynamic binding in Java
- **Composition vs Inheritance in Java**
- Java Inner Classes [Nested Class]
- Java Autoboxing and Unboxing
- Java Wrapper Classes
- Java Ternary Operator
- Java Encapsulation with example
- Java Packages with examples
- Abstract Class in Java
- Interface in Java
- Difference between Abstract Class and Interface in Java
- Garbage Collection in Java
- final keyword In Java

Core Java Tutorial – String Manipulation

- String Comparison
- String immutable
- String Builder
- String Basic
- Why String is immutable and final?
- Understanding Java String Pool
- Java String subsequence example
- Java String compareTo example
- Java String substring example
- Converting String to char and vice versa
- Java Split String example
- String to byte array and vice versa
- String to char array
- Java String concatenation
- String, StringBuffer, and StringBuilder in Java
- String Programs in Java
- Core Java Tutorial – Arrays
- Initializing an Array in Java

- Two-dimensional array in java
- Java Array of ArrayList
- String to String Array Example
- Java Variable Arguments Explained
- Java Array add elements
- Sorting an Array in Java
- Java String Array to String
- Java ArrayList to Array
- Converting Array to ArrayList in Java
- How to copy arrays in Java
- Core Java Tutorial – Annotation and Enum
- Java Annotations Tutorial
- Java @Override Annotation
- Java Enum Example Tutorial

Core Java Tutorial – Collection & Generics

- Java Collections Framework Tutorial
- Java List **<interface>** [ArrayList, LinkedList, Vector & Stack]
- Java Set **<interface>**[HashSet, LinkedHashSet & TreeSet]
- Java SortedSet **<interface>**
- Java NavigableSet **<interface>**
- Java Queue **<interface>** [LinkedList, PriorityQueue & Dequeue]
- Java Map **<interface>** [HashMap, LinkedHashMap, HashTable & TreeMap]
- Java Sorted Map **<interface>**
- Java Navigable Map **<interface>**
- Java ArrayList **<class>**
- Java LinkedList **<class>**
- Vector in Java **<class>**
- Java Stack **<class>**
- Java HashSet **<class>**
- LinkedHashSet In Java **<class>**
- Java TreeSet **<class>**
- Java HashMap **<class>**
- Hashtable in Java **<class>**
- Java LinkedHashMap **<class>**
- Java TreeMap **<class>**
- PriorityQueue **<class>**
- Deque & ArrayDeque **<class>**
- Comparable and Comparator Interface in Java
- Java Iterator & ListIterator
- Java PriorityQueue Example
- ArrayList vs CopyOnWriteArrayList
- How to avoid ConcurrentModificationException when using an Iterator
- Java Generics Example Tutorial

Core Java Tutorial – IO Operations

- Create a New File in Java
- Delete a File in Java
- File separators in Java
- Delete a Directory Recursively in Java
- Rename and Move a File in Java
- Getting File Size in Java
- Get File Extension in Java
- How to check if File exists in Java
- How to check if File is a Directory in Java
- How to get File last modified date in Java
- Java FileNameFilter example to list specific files
- Java File Path, Absolute Path, and Canonical Path Explained
- How to set File Permissions in Java
- 4 ways to copy File in Java
- Reading File in Java using BufferedReader, Scanner, Files
- Java Scanner Class
- Open a File in Java
- Read a File to String in Java
- Java Read file line by line
- How to write a File in Java
- How to append data to a File in Java
- Converting InputStream to File
- Java Random Access File Example
- Download File from URL Example
- Java GZip Example
- Temp Files in Java
- Reading a CSV File using Java Scanner Class
- Java Property File Example

Core Java Tutorial – Exception Handling

- Exception Handling in Java
- Java try-catch block
- Java finally block
- Throw and Throws in Java
- Custom Exception in Java
- java.lang.NoSuchMethodError
- java.lang.NullPointerException

Core Java Tutorial – MultiThreading and Concurrency

- Threads in Java
- Thread Life Cycle
- Thread Sleep Example
- Thread join Example
- Thread wait, notify, notifyAll Example
- Thread Safety in Java – Synchronization
- Java ThreadLocal Example
- Java Timer and TimerTask Example
- Java Thread Pool Example
- Java Callable Future Example

- Java FutureTask Example
- Java ScheduledThreadPoolExecutor Example
- Java Lock Example

Core Java Tutorial – Regular Expressions

- Regular Expressions in Java Example Tutorial
- Validating Email Address using Regular Expression in Java
- Validating Phone Number using Regex in Java
- Core Java Reflection API
- Java Reflection API Tutorial

Java 7

- String in switch case
- Try with Resources – Java ARM
- Binary Literals in Java
- Underscores in Numeric Literals
- Catching Multiple Exceptions in a single catch block
- Java PosixFilePermission example to set File Permissions

Java 8

- Java 8 Features Overview
- Java 8 interface changes
- Lambda Expressions in Java
- Stream API in Java
- Java Date Time API Example Tutorial
- Java Spliterator

Java 9

- Java 9 Features
- Java 9 private method in interfaces
- Java 9 try-with-resources improvements
- Java 9 Optional class improvements
- Java 9 Stream API improvements
- Java 9 “var” for local variables
- Java 9 “_” (underscore) changes
- Java 9 Factory Methods for Immutable List
- Java 9 Factory Methods for Immutable Set
- Java 9 Factory Methods for Immutable Map
- Java 9 Modules
- Java 9 Module Basics Part 2
- Develop Java Module using Command Prompt
- Develop Java Module using Eclipse
- Develop Java Module using IntelliJ IDEA

Java 10

- Java 10 Features
- Java 10 Local Variable Type Inference

Java 11

- Java 11 Features
- 6 New Methods in Java 11 String Class
- Core Java Tutorial – Advanced Topics
- Java Heap Memory and Stack Memory
- Java is Pass by Value and not Pass by Reference
- JVM Memory Model and Garbage Collection
- Serialization in Java
- Java System Class
- Internationalization (i18n) in Java
- Atomic Operations in Java
- Thread Dump in Java
- Deadlocks in Java
- Sorting Objects in Java
- Understanding JDK, JRE, and JVM
- Java Classloader Example Tutorial
- Java clone object

Java Interview Questions

- Java Collection Interview Questions
- Core Java Interview Questions
- Core Java Interview Questions For Freshers
- Basic Java Interview Question

We hope we have covered all the required tutorials for a Java Selenium testers. But if you think that we have missed out on something, please go ahead and comment below. I will write something on that and add that to the list.

What are the core Java concepts?
In Core Java, mostly we have discussed about topics like OOPS concepts, Java constructs like loops and data types, String handling, Collection framework & Exception handling.
Where is core Java used?
Core Java is mainly used to design application software for both desktop and server environments. Core Java is the basic of Java which set the foundation for other editions of the programming language.
What is the difference between core Java and advanced Java?
Java is a general-purpose high-level programming language that helps to build a variety of applications. Core Java is used to build general applications while the Advanced Java is used to build enterprise level applications.
Why is core Java important?
In core java, a java program learner learn the very important basic concepts of java like oops concept, exception handling, collection frmaework. So, every program should be aware those concepts.