CORE JAVA TUTORIAL FOR BEGINNERS

91+8681884318
enquiry@softlogicsys.in
https://www.softlogicsys.in/

Share on your Social Media

Core Java Tutorial for Beginners

Published On: August 7, 2024

Core Java Tutorial for Beginners

Core Java provides the fundamental components needed to create Java programs. In this Core Java tutorial, we explore them in detail with examples.

[Core Java Tutorial PDF]

Introduction to Core Java

The term "Core Java" does not refer to the actual programming language, but rather a set of libraries that have been utilized in application development with Java. In this core Java tutorial, we cover the following:

- Overview of Core Java
- Hello World
- Java Editions

Featured Articles

Want to know more about becoming an expert in IT?

Click Here to Get Started

100% Placement Assurance

AUTHORISED CERTIFICATION PARTNER IBM

Quick Enquiry

Related Courses at SLA

Related Posts

Tableau Developer Salary in Chennai

Published On: October 12, 2024

Introduction A Tableau Developer designs, develops, and maintains dashboards and visualizations using Tableau software. Key...

- Core Java Concepts
- Core Java Terminologies
- Data Types in Java
- Sample Program with Core Java

Overview of Core Java

Core Java is a basic version of the Java programming language that lays the framework for all upcoming Java versions and related technologies such as Java Virtual Machine and CORBA. Java is safe, portable, and does not contain concepts like operator overloading or pointers, making it an easy programming language to learn.

Core Java Interview Questions

Why Core Java?

Programmers can easily become proficient in Java.

- The industry still gains from Java's popularity.
- Most websites and applications in the fields of education, healthcare, government, and defense still make use of Java technology.
- There are several career paths you might take if you decide to pursue a career in Java.

In short, almost everything is possible with Java.

Features of Java

- Object-Oriented Programming Language.
- Platform Independent
- Simple to Learn
- Secured using JVM
- Architecture Neutral
- Robust and Portable
- Multithreading Support.

Applications of Core Java

Core Java concepts can be applied to the following areas:

- Mobile Applications

- Desktop GUI Applications
- Artificial intelligence
- Scientific Applications
- Cloud Applications
- Embedded Systems
- Gaming Applications

Java has many features and several libraries that you can use to accomplish almost anything. Any network or the internet can be used to distribute the Java application.

```
Hello World Program in Java

class Simple{

    public static void main(String args[]){

    System.out.println("Hello World");

    }

}
```

Download Core Java Syllabus PDF

Java Editions

Let's now examine the various Java platform editions:

Java SE: Java Standard Edition

A computer platform for creating desktop or window-based apps is Java SE. The foundational portion of Java SE, known as Core Java, is where programmers use the fundamental ideas of Java to create desktop apps. The Java Development Kit, or JDK, is a popular Java SE implementation.

Java EE: Java Enterprise Edition

This edition, or Java 2 Platform, Java Enterprise Edition, or J2EE, is geared toward enterprise development and allows programmers to create server-side applications. A popular tool for web development is Java EE.

Java ME: Java Micro Edition

Java ME is a micro version designed for application development on mobile devices. It is designed especially for creating mobile applications. Additionally, it facilitates the operation of Java-based apps on mobile devices with constrained RAM and processing capacity.

Core Java Concepts

Generally, there are many components in core Java. They are as follows:

Core Java Fundamentals

It stands for the primary version of Core Java, which is usually utilized for creating standard desktop programs.

It includes a portion of Java SE technology, including general-purpose and specific-purpose APIs, and combines fundamental Java features with a thorough command of the language.

OOPs Concepts

The programming paradigm known as object-oriented programming, or OOP, is primarily built on the ideas of objects, which involve data and behavior.

Here are the elements of OOP concepts:

- **Class:** These are the templates that dictate how the items behave and are put together. It includes both data and the functions based on such data. A class declaration in Java may contain the following components:
    - **Modifiers:** A class may have a default access level class keyword, be private or public, or both. The class keyword is used to create a class.
    - **Class keyword:** A capital letter should typically appear at the beginning of the class name.

- - **Superclass (optional):** We use the extends keyword and follow the class name with the name of the superclass if the class has one.
  - **Interface (optional):** If a class implements an interface, the class name comes first, followed by the implements keyword and the interface's name.
- **Object:** These are essentially real-world examples of classes that illustrate the structure. Three characteristics are present in an object:
  - **State:** An object's state serves as a representation of its data, or value.
  - **Behavior:** The term "behavior" refers to an object's functioning, such as deposit, withdrawal, and so forth.
  - **Identity:** An object's identity is frequently represented with a unique ID. The external user cannot see the ID's value. The JVM uses it internally to give each object a distinct identity.
- **Encapsulation:** It is the process of collecting data and the way a method operates on the data that is part of a class.
- **Inheritance:** The ability to inherit traits and behaviors from other classes, also referred to as superclasses, helps the class.
- **Polymorphism:** The ability to consider objects of different class types as belonging to the superclass is known as polymorphism.
- **Abstraction:** It focuses mostly on hiding details about how objects are implemented and only displays key components.

Overloading and Overriding

Method Overloading: It allows many methods with similar sorts of names to be included in a class of different types of parameter lists.

Method Overriding: It's the process by which a subclass carries out an implementation of a method that differs from what its superclass

specifies.

Exception Handling

Exceptional handling helps in the handling of unexpected events and runtime failures by program developers. The "try-catch blocks" are utilized to handle and catch exceptions. Thus, abrupt program termination is avoided.

Packages

The fundamental ideas of Java are that packages enable the arrangement and clustering of related classes and interfaces. This allows you to maintain the namespace and permits code reuse and maintenance.

Collections

Java collections are used to manage groups of things by providing an interface and class for storing and manipulating various objects and their groups. The primary data structures that offer various algorithms to associate with such a collection are lists, maps, sets, and so on.

Multithreading

Java multithreading enables the simultaneous operation of several threads. Multithreading is supported natively via the Thread class and the Runnable interface in Core Java.

Swings

A graphical user interface (GUI) in the Core Java concept called Swings aids in the creation of high-quality desktop applications.

It has a specific set of elements, such as labels, text fields, and buttons. These capabilities can be tailored to create interactive apps.

Applets

Applets are little Java programs inserted in web pages that were formerly praised as dynamic

online content. Applets used to be a major component of websites' interactive content, but as web technologies have advanced, their usage has decreased.

JDBC

Java Database Connectivity, or JDBC for short, is a core technology that lets Java programs communicate with databases.

JDBC is a crucial component of many Java programs that include data storage because it allows developers to carry out operations including data entry, retrieval, and modification.

Core Java Developer Salary

Core Java Terminologies

Here are the popular Java terminologies:

AWT (Abstract Window Toolkit): A collection of classes called AWT is used to create applets and standalone program components, including buttons, menus, and scrollbars.

API (Application Programming Interface): The description of how an application programmer can access the state and behaviour of classes and objects.

Array: An integer is used to uniquely identify each point in this collection of identical data objects.

Bytecode: The JDK's Javac compiler transforms the Java source code into bytecode so that the JVM may execute it.

DOM (Document Object Model): A W3C specification-defined tree of objects with interfaces for producing an XML representation of the tree and navigating around it.

Garbage Collector: The JVM provides a program

named Garbage Collector that can be used to remove or retrieve that memory.

JVM (Java Virtual Machine): It runs the bytecode that the compiler produces.

JDK (Java Development Kit): This all-inclusive set of tools for developing Java includes the compiler, Java Runtime Environment (JRE), debuggers, documentation, and more.

JRE: It does not allow us to compile the Java program, but it does allow it to run.

JMF (Java Media Framework): Clocks that synchronize various media (such as audio and visual output) are supported under the core framework. One may perform full audio and video streaming with the standard extension architecture.

Finalize Method: The trash collector calls the finalize method right before an object is destroyed or erased.

Constructors in Java

A constructor in Java is a code block that resembles a method. The constructor is invoked each time a new instance of the class is created. The constructor must be called in order for the object's memory to be allocated.

Java has two different kinds of constructors. They are listed in the following order:

Default Constructor: A constructor type that doesn't require any parameters is called a default constructor. The compiler automatically creates a default constructor for a class without any arguments when we forget to define one.

Parameterized Constructor: These constructors are known as parameterized constructors. During initialization, it is used to set custom values for a class's fields.

Keywords in Java

Reserved words are also referred to as keywords in Java. These are specific terms with distinct connotations. 61 specified reserved keywords in Java cannot be used as variable, object, or class names. The following is a list of keywords in Java:

abstract	boolean	protected	void	break
byte	catch	return	with	case
do	double	short	try	enum
float	int	this	vloatile	interface
class	private	while	to	public
throws	switch	super	static	main
import	for	if		

Data Types in Java

The various sizes and values that can be stored in a variable based on requirements are known as data types in Java.

There are two more varieties of Java Datatype:

Primitive Data Type in Java

Primitive data types in Java provide the framework for data manipulation. These represent the most fundamental categories of data used by the Java programming language. There are numerous primitive data types in Java, such as:

boolean: 1 bit size, returns true or false.

byte: 8 bits, -128 to 128.

char: 16 bits, 0 to 255 and ASCII values.

short: 16 bits, -32,768 to 32,768.

int: 32 bits, -2,147,483,648 to 2,147,483,647

long: 64 bits, -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807

float: 32 bits, up to 7 decimal digits.

double: 64 bits, up to 16 decimal digits.

Non-Premitive Data Type

Primitive datatypes are the building blocks for non-primitive datatypes. A few instances of non-primitive datatypes are queues, stacks, and arrays.

Variables in Java

The storage units for data values are called variables. Every variable is assigned based on the type of data to which it belongs.

Syntax

data_type var_name;

As listed below, there are three different kinds of data types in Java:

Local Variables: A local variable is defined inside a constructor, method, or block.

Instance Variables: Non-static variables called instance variables are defined outside of constructors, methods, and blocks in a class.

Static variables: Class variables are another name for static variables. Within a class, the static keyword is used to define the static variables outside of any constructors, methods, or blocks.

Core Java Training

Comments in Java

In Java, there are three categories for comments.

Single Line Comment

You can use a double forward slash "//" followed by your message to remark on a single line of code. For coding, this syntax is frequently utilized.

*/*single line comment*/*

Multi-line Comment

A double forward slash "/*" can be used in syntax to remark on multiple lines of code. All you have to do is type your message in between the two symbols and end the comment with "*/". In coding, this syntax is often utilized.

/ These lines are*

ignored by

*compiler */*

JavaDoc Type Comment

Using this strategy is generally beneficial while working on a project or software package, since it can help create a reference documentation page.

*/**Comment starts*

*This is

sample comment/

Sample Program with Core Java

Finding the Longest Repeating Sequence in a String

```
public class LongestRepeatingSequence

public static void main(String[] args) {

String str = "aabcaabdaab";

System.out.println("Longest repeating sequence: " +
longestRepeatingSequence(str));

}
```

```java
public static String longestRepeatingSequence(String str) {

int n = str.length();

int[][] dp = new int[n + 1][n + 1];

int longest = 0, endIndex = 0;

for (int i = 1; i <= n; i++) {

for (int j = i + 1; j <= n; j++) {

if (str.charAt(i − 1) == str.charAt(j − 1) && dp[i − 1][j − 1] < (j − i)) {

dp[i][j] = dp[i − 1][j − 1] + 1;

if (dp[i][j] > longest) {

longest = dp[i][j];

endIndex = i;

}

} else {

dp[i][j] = 0;

}

}

}

return str.substring(endIndex − longest, endIndex);

}

}
```

Conclusion

This Core Java tutorial will be helpful to you for understanding the basics of the Java programming language. Gain expertise with our **Core Java training in Chennai** for a promising career in software development.

Share on your Social Media

Softlogic Academy

Softlogic Systems

KK Nagar [Corporate Office]

No.10, PT Rajan Salai, K.K. Nagar, Chennai
– 600 078.
Landmark: Karnataka Bank Building
Phone: +91 86818 84318
Email: enquiry@softlogicsys.in
Map: Google Maps Link

OMR

No. E1-A10, RTS Food Street
92, Rajiv Gandhi Salai (OMR),
Navalur, Chennai - 600 130.
Landmark: Adj. to AGS Cinemas
Phone: +91 89256 88858
Email: info@softlogicsys.in
Map: Google Maps Link

Courses

- Python
- Software Testing
- Full Stack Developer
- Java
- Power BI
- Clinical SAS
- Data Science
- Embedded
- Cloud Computing
- Hardware and Networking
- VBA Macros
- Mobile App Development

Important Links

- Disclaimer
- Privacy Policy
- Terms and Conditions

Social Media Links

Review Sources

- Google
- Trustpilot
- Glassdoor
- Mouthshut
- Sulekha
- Justdial
- Ambitionbox
- Indeed

DevOps

Software Suggest

Sitejabber