Java is among the top[**computer languages**](https://leverageedu.com/blog/computer-languages/) used for web, mobile, desktop applications, etc. It is a programming language based on the concept of Object-Oriented Programming Systems (OOPS). Programming enthusiasts can learn Java to upscale their skills and knowledge. After the entire Java Web application directory structure. Let’s explore what is included in the advanced, framework and core java syllabus and curriculum.

**What is Core Java?**

The term “core” refers to the fundamental notion of anything, while the phrase “Core Java” refers to the fundamental concept of the Java programming language. We are all aware that Java is one of the most well-known and frequently used programming languages, and that a novice should begin with Core Java and work their way up to Advance Java. Java programming language is a general-purpose programming language based on the object-oriented programming (OOPs) paradigm. Java’s ocean is too deep to learn, i.e., the more you study, the deeper it becomes. Java is a powerful and platform-independent programming language. Java follows the WORA philosophy, which stands for Write Once, Run Anywhere. The programming language is straightforward and simple to grasp. However, it is important to note that Core Java is not the same as Java. Although Java is self-contained, it is normal for newcomers to start with the fundamentals of the language. In reality, there are several editions of Java, with Core Java being one of them.

**Java Editions**

The Java Programming Language supports the following declared editions:

|  |  |
| --- | --- |
| Java SE (Java Standard Edition) | Java SE is a computer platform that may be used to create desktop or Windows-based applications. Thus, core Java is a portion of Java SE in which developers create desktop-based programs utilizing Java’s fundamental ideas, with JDK (Java Development Kit) being a well-known Java SE implementation. |
| Java EE (Java Enterprise Edition) | Also known as J2EE or Java 2 Platform. It is the enterprise platform on which a developer creates applications for servers, i.e. enterprise development. This is the edition for web development. |
| Java ME (Java Micro Edition) | It is the tiny version that is used for mobile phone application development. As a result, Java ME is required for the creation of mobile apps. As a result, Core Java is clearly a part of Java SE, and Java SE serves as the foundation for all subsequent Java versions. |

**Types of Java Courses**

Java is categorized into 3 types with very minor differentiation among them.

* Core Java (Java SE & JDBC)
* **Advanced Java** (Servlets and JSP)
* Different types of FrameWorks (Spring & Hibernate etc.)

Credits: IBT Training

**Concepts Covered in Core Java**

**The following are some of the most important Java fundamental topics for a novice to learn:**

* Java Fundamentals
* OOPs Concepts
* Overloading & Overriding
* Inheritance with Interface and Abstract Class
* Exception Handling
* Packages
* Collections
* Multithreading
* Swings
* Applets
* JDBC (Basic Database Connections)

**Core Java Syllabus**

Professionals and students who want to get themselves certified in Core Java or JDBC can refer to this syllabus for learning and enhancing their knowledge of Java during their academic sessions. Here is more detailed information about the core java syllabus:

**Basic of Java**

* What is Java?
* History and Features of Java
* C++ vs. Java
* Hello Java Program
* Internal How to set the path?
* JDK, JRE, and JVM (Java Virtual Machine)
* JVM Memory Management
* Internal Details of JVM
* Unicode System, Operators, Keywords, and Control Statements like if-else, switch, for loop, and while loop

**Class, Objects, and Types of Classes**

* Naming conventions of Java
* Classes, objects, and features
* Object declaration and initialization
* The life cycle of an object
* Anonymous object in Java

**Packages in Java**

* How to declare a package in a company project
* Package naming conventions
* Sub packages
* Types of packages such as user-defined packages, built-in packages
* Importing packages in Java

**Data Types in Java**

* Data types in Java
* Primitive data types
* Non-primitive data types
* Memory allocation of primitive and non-primitive data types

**Variables, Constraints, and Literals**

* Variable declaration and initialization
* Naming convention
* Types of variables such as local variables, instance variables, and static variables
* Scope and memory allocation of variables

**Methods in Java**

* Methods in Java
* Use of method in Java
* Method declaration, the method signature
* Types of methods in Java: predefined method, user-defined methods: instance method, static method
* Calling of method
* Java main method
* Return type in Java

**Constructor in Java**

* What is Constructor in Java?
* Types of Constructors: Default and parameterized constructors
* Java constructor overloading
* Constructor chaining in java
* Copy constructor in Java

**Modifiers in Java**

* What are an Access modifier ad a non-access modifier in Java?
* Types of access modifiers like private, default, protected, and public
* Types of non-access modifiers like abstract, final, native, static, strictfp, synchronized modifier, transient, and volatile.

**Static Keyword**

* What is Static Keyword
* Static variable
* Static method
* Static block, instance block
* Static Nested Class in Java
* Difference between static variable and instance variable, static method, an instance method, static block, and instance block

**Final Keyword**

* Final keyword
* Final variable
* Final method
* Final class

**Inner Class in Java**

* What is Inner Class in Java?
* Types of Inner class in Java

**Super and this keyword**

* Super Keyword
* Calling of superclass instance variable
* Superclass constructor
* Superclass method

**Encapsulation**

* Encapsulation in Java
* How to achieve encapsulation
* Data Hiding
* Tightly encapsulated class
* Getter and setter method in Java
* Naming convention of getter and setter method

**Inheritance**

* Inheritance in Java
* Is-A-Relationship
* Aggregation and Composition
* Types of Inheritance

**Polymorphism**

* Polymorphism in Java
* Types of Polymorphism
* Static and Dynamic Binding
* Method overloading]
* Method Overriding

**Abstraction**

* Abstraction in Java
* Abstract Class
* Abstract method
* Interface in Java
* Nested interface, rules, and example programs

**Complete Industrial Core Java Syllabus and Sub Topics**

There are 31 chapters in this curriculum. The following is the title of the chapter, along with a brief description:

**1. Basics of Java**

The first chapter covers the fundamentals of the Java programming language, such as

* What is Java?
* History and Features of Java
* C++ vs Java
* Hello Java Program
* internal How to set the path?
* JDK, JRE, and JVM (Java Virtual Machine)
* JVM Memory Management
* Internal details of JVM
* Unicode System, Operators, Keywords, and Control Statements like if-else, switch, For loop, while loop, etc.

**2. Class, Object, and Types of classes**

The most fundamental and core ideas of Java are covered in Chapter 2. They are as follows:

* Naming convention of Java
* Classes, Objects, and Features. It explains how to declare a class, and how to create an object in Java.
* Object declaration and initialization
* Life cycle of an object
* Anonymous object in Java

**3. Packages in Java**

Packages in Java are covered in Chapter 3. The following topics will be covered in this chapter.

* How to declare a package in a company project
* Package naming conventions
* Sub packages
* Types of packages such as user-defined packages, built-in packages
* Importing packages in Java

**4. Data types in Java**

The following Java concepts are covered in this chapter.

* Data types in Java
* Primitive data types
* Non-primitive data types
* Memory allocation of primitive and non-primitive data types, etc.

**5. Variables, Constants, and Literals**

Variables, constants, and literals are the three topics covered in Chapter 5. This chapter will cover the following subtopics.

* Variable declaration & initialization
* Naming convention
* Types of variables such as local variables, instance variables, and static variables
* Scope and memory allocation of variables.

**6. Methods in Java**

* Methods in Java
* Use of method in Java
* Method declaration, method signature
* Types of methods in Java: predefined method, user-defined methods: instance method, static method
* Calling of method
* Java main method
* Return type in Java.

**7. Constructor in Java**

You will learn about the following things in this chapter:

* What is Constructor in Java?
* Types of constructors: Default and Parameterized constructors
* Java constructor overloading
* Constructor chaining in java
* Copy constructor in Java

**8. Modifiers in Java**

This chapter covers Topics such as

* In Java, what is the difference between an access modifier and a non-access modifier?
* Private, default, protected, and public are examples of access modifiers.
* Types of Non-access modifiers like abstract, final, native, static, Strictfp, synchronized modifier, transient, volatile.

**9. Static Keyword**

The following significant Topics are covered in this chapter.

* What is Static keyword?
* Static variable
* Static method
* Static block, Instance block
* Static Nested Class in Java
* Difference between static variable and instance variable, static method and instance method, static block, and instance block.

**10. Final Keyword**

The following significant Topics are covered in this chapter.

* Final keyword
* Final variable
* Final method
* Final class.

**11. Inner Class in Java**

The following significant Topics are covered in this chapter.

* What is Inner class in Java?, Properties of the inner class, Instantiating inner class.
* Types of inner class in Java: Normal inner class, Method local inner class, Anonymous inner class, and Static nested class.

**12. Super and this Keyword**

The following significant Topics are covered in this chapter.

* Super keyword
* Calling of superclass instance variable
* Superclass constructor
* Superclass method.
* The second section deals with
* This keyword
* Calling of current class constructor, and method.

**13. OOPs concepts**

The most essential topic in this chapter is the object-oriented programming system (OOPs). You will learn about class, object, encapsulation, inheritance, polymorphism, and abstraction as part of the OOPs paradigm. For the objectives of the interview, all of the issues are quite crucial.

**14. Encapsulation**

The following significant Topics are covered in this chapter.

* Encapsulation in Java
* How to achieve Encapsulation
* Data hiding
* Tightly encapsulated class
* Getter and setter method in Java
* Naming convention of getter and setter method

**15. Inheritance**

The following significant Topics are covered in this chapter.

* Inheritance in Java
* Is-A Relationship
* Aggregation and Composition(HAS-A)
* Types of inheritance:  Single level, Multilevel, Hierarchical, Multiple, and Hybrid inheritance.

**16. Polymorphism**

The following significant Topics are covered in this chapter.

* Polymorphism in Java,
* Types of polymorphism: Compile-time polymorphism and Run-time polymorphism
* Static and Dynamic Binding
* Method overloading
* Method overriding
* Rules of method overloading and method overriding, various example programs related to rules of overloading and overriding.
* Covariant Return type

**17. Abstraction**

The following significant Topics are covered in this chapter.

* Abstraction in Java
* Abstract class
* Abstract method
* Interface in Java
* Nested interface, rules, and example programs.

**18. Garbage Collection**

In this chapter, you’ll learn about garbage collection in Java.

**19. Input-Output Stream**

The following significant Topics are covered in this chapter.

* FileOutputStream, FileInputStream
* BufferedOutputStream, BufferedInputStream
* SequenceInputStream
* ByteArrayOutputStream, ByteArrayInputStream
* DataOutputStream, DataInputStream
* Java FilterOutputStream, Java FilterInputStream
* Java ObjectStream, Java ObjectStreamField
* Console
* FilePermissionWriter, Reader, FileWriter, FileReader
* BufferedWriter, BufferedReader
* CharArrayReader, CharArrayWriter
* PrintStream, PrintWriter
* OutputStreamWriter, InputStreamReader
* PushbackInputStream, PushbackReader
* StringWriter, StringReader
* PipedWriter, PipedReader
* FilterWriter, FilterReader, File FileDescriptor, RandomAccessFile, and  java.util.Scanner.

**20. Collections Framework**

The following significant Topics are covered in this chapter.

* What is Collections Framework?
* List, Set, SortedSet, Queue, Deque, Map, Iterator, ListIterator, and Enumeration.
* ArrayList, LinkedList, HashSet, LinkedHashSet, TreeSet, ArrayDeque, PriorityDeque, EnumSet, AbstractCollection, AbstractList, AbstractQueue, AbstractSet, and AbstractSequentialList.
* Map, Map Entry, SortedMap, and NavigableMap
* HashMap, LinkedHashMap, TreeMap, IdentityHashMap, WeakHashMap, and EnumMap.
* Comparator, RandomAccess interfaces as well as Observable class.

**21. Serialization**

Serialization, Deserialization, and the Java temporary keyword are all covered in this chapter.

**22. Exception Handling in Java**

This chapter is extremely necessary for any Java technical exam or interview. This chapter will introduce you to

* Exception Handling in Java
* Try-catch block
* Multiple Catch Block
* Nested try block
* Finally block
* Throw Keyword
* Throws Keyword
* Throw vs Throws, Final vs Finally vs Finalize
* Exception Handling with Method Overriding Java Custom Exceptions

**23. Java Annotations**

This chapter deals with Java annotations, Built-In Java annotations like @Override, @SuppressWarnings, @Deprecated, @Target, @Retention, @Inherited, @Documented, Java custom annotations, and types of annotations.

**24. Reflection in Java**

* Reflection API
* NewInstance() & Determining the class object
* Javap tool, Creating javap tool
* Creating applet viewer
* Accessing private method from outside the class

**25. Java Array**

This chapter is about

* Java Array
* Types of array: single dimensional array, multidimensional array, declaration, instantiation, and initialization of Java array
* Passing array to a method
* Anonymous array in Java
* Cloning an array in Java

**26. String, String Buffer, String Builder**

This is the most crucial chapter in the whole Java core curriculum. It will primarily cover three subjects, namely

* String,
* Immutable String
* String Comparision, String concatenation
* Substring
* StringBuffer class
* StringBuilder class
* To String method
* StringTokenizer class

**27. Java Thread**

* Java multithreading
* Multithreading life cycle of a thread creating
* Thread scheduler
* Sleeping a thread, Start a thread twice
* Calling run() method
* Joining a thread
* Naming a thread
* Thread priority,
* Daemon thread
* Thread pool
* Thread group
* Shutdownhook
* Java Synchronization: synchronized method, synchronized block, static synchronization
* Deadlock
* Inter-thread Communication
* Interrupting Thread

**28. JDBC**

This chapter is about

* JDBC Drivers
* Steps to connect to Database
* Connectivity with Oracle
* Connectivity with MySQL
* Connectivity with Access without DSN
* DriverManager
* Types of JDBC statements: Statement, Prepared statement, Callable statement
* Database Metadata, Resultset Metadata
* ResultSet, types of ResultSet,
* Storing image, Retrieving image
* Storing file, Retrieving file, Stored procedures, and functions
* Transaction Management
* Batch Processing
* JDBC New Features, Mini Project, and interview questions.

**29. Agile**

This chapter is about

* Agile model
* Advantages, and Disadvantages of Agile model
* Agile versus Waterfall method
* Important terminology: Scrum, Scrum Master, Flow of Agile Implementation, Sprint, and Burn down Charts.

**30. Design Pattern**

This chapter is about

* Singleton Object
* Singleton design pattern with Serialization
* Factory Pattern
* Abstract Factory.

**31. Database**

**Core Java Vs. Advance Java**

|  |  |
| --- | --- |
| **Core Java** | **Advance Java** |
| The Java programming language’s fundamental fundamentals are covered in Core Java. | The advanced themes and principles of the Java programming language are covered in Advance Java. |
| Core Java is a programming language that is used to create computing and desktop apps. | Advance Java is a programming language that is used to create corporate applications. |
| To begin with, Java is the first step. | Following the completion of Core Java, this is the next phase. |
| The architecture of Core Java is single-tier. | The architecture of Advance Java is two-tiered. |
| It’s classified as Java SE. | It’s classified as Java EE or J2EE. |
| It covers fundamental concepts like OOPs, inheritance, and exception handling, among others. | JDBC, servlets, JSP, web services, and other advanced topics are covered. |

**Also Read:** [**Difference Between C and JAVA**](https://leverageedu.com/blog/difference-between-c-and-java/)

**Different Types of Core Java Frameworks**

This section will show you some of the most popular Java frameworks of 2020. While there are more than just 10 such frameworks, the ones listed and discussed stick out because of their significant features and ease of use.

**Spring**

Spring is a core part of the java syllabus and is a very lightweight implementation of the framework, usable for pretty much any type of Java project. It’s a modular framework that you could use for any level or layer of a project. What makes it stick out is the fact that you can use it to work on not just one layer of a project but also the entire scope.

**Hibernate**

Hibernate is an object-relational mapping (ORM) framework that makes common data-handling mismatch cases a thing of the past. If you’re always working with relational databases, the Hibernate ORM framework could easily become your staple.

The framework comes stock with data handling muscle that bridges paradigm differences. Companies like IBM and Dell have used the Hibernate framework for their web applications.

**JSF (JavaServer Faces)**

It’s often a huge task for back-end developers to get the front side of complex applications right. This is where JSF comes in handy.

The Oracle-built, stable framework comes with a component-based MVC environment to create beautiful “faces” for Java applications. It’s packed to the brim with libraries that allow developers to experiment with the front end—without introducing other frameworks for that part.

**GWT (Google Web Toolkit)**

As can be expected from a Google product, GWT is open source. The main reason many developers’ work starts with GWT is that it’s easy to make beautiful UIs with little knowledge of front-end scripting languages. It basically turns Java code into browser-friendly packages.

Web apps such as Blogger, Google Analytics, and Google Adsense are all built using Java with the GWT framework. It’s fully featured and supported by a large group of developers dedicated to the framework, making it perfect for scale-sensitive application development.

**Struts (The Later Version)**

Strut is also part of the core java syllabus and is an Apache-run enterprise-level framework perfect for web developers. It’s feature-rich and comes in two versions: Struts 1 and 2. The most widely used is Struts 2, which basically extends the first version with everything that comes with OpenSymphony web framework tools. That means you get to apply new technologies such as Ruby and new JavaScript frameworks to extend your Java applications’ functionality.

**FAQs**

**Q1. What are the important concepts in core Java?**

Ans. Concepts from core Java:
**OOPS, concepts (Data Abstraction, Encapsulation, Inheritance, Polymorphism)** Basic Java constructs like loops and data types. String handling. Collection framework.

**Q2. What are the 3 components of Java?**

Ans. There are three main components of Java language: JVM, JRE, and JDK. Java Virtual Machine, Java Runtime Environment, and Java Development Kit respectively.

**Q3. Is core Java in demand?**

Ans. Java is one of the most popular programming languages, and companies big and small use it in their tech stacks thanks to its speed and reliability. Check Uber, Netflix, Spotify, Google, and Amazon – each of these household names uses Java. Hence, **Java developers are in high demand**

**Q4. What is Advance Java?**

Ans. Advanced Java is a programming language that is used to create corporate applications.

**Q5. What are the concepts covered in Core Java?**

Ans. Java Fundamentals
OOPs Concepts
Overloading & Overriding
Inheritance with Interface and Abstract Class
Exception Handling
Packages
Collections
Multithreading

There is no certain limit to what you can develop and discover by learning a new programming language. Which course you are planning to pursue in the upcoming intake? Let us know in the comment section below. Stay tuned to [**Leverage Edu**](https://leverageedu.com/) for the latest updates around the trending courses.

**Share this article**

**Leave a Reply**

Top of Form

Your contact details will not be published. Required fields are marked \*



SUBMIT

Bottom of Form

1. **Akanksha dongare**

[May 13, 2022 at 1:02 pm](https://leverageedu.com/blog/core-java-syllabus/#comment-421324)

Awesome website u have created …I really appreciate your work

[**Reply**](https://leverageedu.com/blog/core-java-syllabus/#comment-421324)

1. **Soniya Vaidya**

[July 5, 2023 at 9:56 am](https://leverageedu.com/blog/core-java-syllabus/#comment-512775)

Nice work really helped to get known about syllabus what actually is the content. Thank you…

[**Reply**](https://leverageedu.com/blog/core-java-syllabus/#comment-512775)

VIEW COMMENTS (2)

** 20,000+ students realised their study abroad dream with us. Take the first step today.**