**Core Java Syllabus:**

Java Language, OOPS, Programming

* Introduction to Java and OOPS
* Java Tokens- Comments, Identifiers, Keywords
* Working with Java Editor Software – Eclipse
* Packages with static imports
* Modifiers
* Datatypes, Literals, Variables, Type Conversion, Casting
* Reading runtime values from keyboard
* Control Statements / Flow control statement
* Method and Types of methods
* Variable and Types of Variables
* Constructor and Types of constructors
* Block and Types of Blocks
* Declarations, Invocations and Executions
* Compiler & JVM Architecture with Reflection API
* Static Members and their execution control flow
* Non-Static Members and their execution control flow
* Final Variables and their rules
* Classes and Types of classes
* OOPS- Fundamentals, Models, Relations and Principles
* Coupling and Cohesion (MVC and LCRP Architectures)
* Types of objects & Garbage Collection
* Arrays

Java API and Project

* API & API Documentation
* Fundamental Classes – Object, Class, System, Runtime
* String Handling
* Exception Handling and Assertions
* Multithreading with JVM Architecture
* IO Streams (File IO)
* Wrapper Classes with Auto boxing & unboxing
* Collections with Generics
* Java 7, 8 new features
* Inner classes
* Formatting date, time (java.text package)

**More detailed syllabus is as below.**

Introduction to Java

* Features of Java
* Introduction to difference java versions
* Introduction to JVM, JIT, class loader and byte code
* JDK, JVM, JRE

Language Fundamentals

* Identifiers and Keywords, Code Comments
* Primitives Data types and non-primitive data types
* Literals and Variables (local, instance, global)
* Flow Control Statements
* Arrays

Operators and Assignments

* Introduction to Operators
* Assignments and Initializations
* Arithmetic operators
* Logical operators

Declarations and Access Control

* Access Modifiers for Class
* Access Modifiers for Class Members
* Packages
* Static imports

Object Oriented Programming

* Creating Classes and Instances
* Inheritance
* Polymorphism (Method Overloading & Overriding) Abstract classes and Interfaces
* Constructors and Initialization
* Reference Variable Casting
* Static data and methods

Object Lifetime

* Object creation and initialization
* Memory Organization
* Garbage Collection
* Object Finalization

Exception Handling and Assertions

* Traditional Error Handling Techniques
* Importance of Exception Handling
* Exception Handling Framework
* Stack-based Execution
* Checked and Unchecked Exceptions
* User defined Exceptions
* Cautions When Using Exceptions
* Assertions
* Exception Chaining

Streams and Files java IO

* Files and Streams
* File handling and File IO
* Serialization

Multithreading

* Java Thread Model
* Thread Class & Runnable Interface
* Types of Threads - User, Demon
* Creating Your Own Threads
* Threads States and Life cycle
* Thread Methods: yield(), sleep(), currentThread() etc
* Scheduling and Thread Priorities
* Concurrency Control and Synchronization
* Deadlocks
* Inter-thread communication

Fundamental Class

* Wrapper Classes
* Autoboxing/ Unboxing
* Math Class
* String and StringBuffer, String Builder Classes
* Date and Calendar

Collections and Maps

* Core Interfaces and classes
* Use of Sets
* Use of Lists
* Use of Maps
* Sorted Maps, Sorted Sets
* Choosing appropriate collections
* Enumerations and Iterators
* Use of Equals and Hashcode methods
* Comparable and Comparator interfaces
* java.util.Collections class
* Generics: Use of generics

Introduction to Inner Classes

* Static/Non-Static Member Class
* Local Class
* Anonymous Class
* Problems with Inner Classes

JDBC

* Types of Drivers
* Database Connectivity
* Executing SQL statements(DDL, DML, DLL)
* Prepared Statements
* Types of Resultsets
* Connection Pooling

Logging

* Fundamentals of Logging
* Understanding different logging levels
* Creating and writing to log files

More Features

* Enums
* Introduction to Annotations