Table of Contents

[1 JAVA COURSES 2](#_Toc527322547)

[1.1 Basic 2](#_Toc527322548)

[1.1.1 About the Trainer 2](#_Toc527322549)

[1.1.2 Introduction 2](#_Toc527322550)

[1.1.3 Course Contents 3](#_Toc527322551)

# JAVA COURSES

## Basic

### About the Trainer

|  |
| --- |
| The basic course in Java is being conducted by an experienced software professional who has more than two decades of experience in the IT industry.  The trainer has extensive and hands on experience in the building of enterprise applications in Java/J2EE (Struts, Spring, Hibernate) and other open source technologies like SOLR, Kafka, Redis, MongoDB for clients across the globe.  Given his strong background in programming, managing and mentoring developers, students participating in the course will benefit from his experience, insights and ability to relate features of Java to solving real world problems.  Using a friendly interactive style, students will be exposed to real code from day one and will teach concepts using a hands on approach. |

### Introduction

|  |
| --- |
| Java is a programming language that has been around since the 1990s.  From the very beginning, Java has been designed to be portable, modular, efficient and extensible.  Both, additional functionality and advanced functionality can be easily incorporated through the large standard library that Java itself provides along with third party libraries developed by the Java community across the globe.  An entire ecosystem has grown up around Java and, over the years, it used for enterprise web application development, application integration, building web services, powering search engines, machine learning, data sciences and other kinds of applications.  As the saying goes – “The proof of the pudding is in the eating” – Leading open source products like Apache SOLR, apache Kafka, ElasticSearch, Apache Spark, Spring Framework, Hibernate, Scala have been developed using the Java ecosystem.  Enterprises, both big and small, have used Java to solve their business problems.  This course requires between 45 to 55 hours and includes approximately 40 examples, 20 exercises that the students will have to complete on their own and 2-4 activities that they will have to execute in groups to foster the concept of collaboration.  Using a code based interactive approach, as opposed to theory first, a more effective learning environment will be provided. From day one, we will dive straight into code !.  We are confident about making the students excited about this great language, curious about exploring features on their own and becoming part of the vibrant Java community !.  **Next Steps -** After completing this course, the students can immediately move onto more advanced topics |

### Course Contents

| **#** | **TOPIC** | **SUB TOPICS** |
| --- | --- | --- |
| 1 | General Introduction | * Informal introduction to Java * Installation instructions * The Java Ecosystem * Widely used IDEs (Eclipse, NetBeans, IntelliJ) * Some sample programs * Reading material that students can refer to for further study |
| 2 | Code – Compile – Run  *(Getting your feet wet in the wonderful world of Java)* | * Create source code file, compile it using the javac compiler, then run the compiled bytecode on a Java virtual machine * Looping constructs * Conditional branching * Data types   + Primitive data types   + Arrays * Exception handling |
| 3 | Classes and Objects  *(Classes are at the heart of Java. We dive straight into the best feature that Java has to offer and explain the key concepts related to Classes and Objects like Inheritance, Polymorphism and others)* | * Class Definitions * Methods * Access control * Understanding “Static” * Constructors * Inheritance * Abstract classes * Interfaces * Final classes |
| 4 | Using the Java Standard Library  *(Java’s standard library is huge. These are pre-built classes that developers can use for implementing various kinds of features. The course will focus on one of the most importance features that the standard library provides – Collections)* | * Collections   + ArrayList, LinkedList, etc. * Working with MAps * String Handling |
| 5 | Generics  *(This feature, widely used across the Java ecosystem enables developers to create parametrized types)* | * Introduction * Examples * Practical use cases of generics |