**Core Java**

**Day 1**

**Java Platform Overview**

* Defining how the Java language achieves platform independence
* Defining how the Java language continues to evolve

**Java Syntax and Class Review**

* Creating simple Java classes
* Using Java fields, constructors, and methods

**Encapsulation and Subclassing**

* Using encapsulation in Java class design
* Modeling business problems using Java classes
* Creating and use Java subclasses
* Using access levels: private, protected, default, and public.
* Overloading methods

Day 2

**Overriding Methods, Polymorphism, and Static Classes**

* Overriding methods
* Using upward and downward casts
* Modeling business problems by using the static keyword

**Abstract and Nested Classes**

* Designing general-purpose base classes by using abstract classes
* Constructing abstract Java classes and subclasses
* Applying final keyword in Java
* Distinguish between top-level and nested classes

**Interfaces**

* Defining a Java interface
* Choosing between interface inheritance and class inheritance
* Extending an interface

**Day 3**

**Collections and Generics**

* Creating a custom generic class
* Using the type inference diamond to create an object
* Implementing an ArrayList

**Exceptions**

* Defining the purpose of Java exceptions
* Using the try and throw statements
* Using the catch, multi-catch, and finally clauses
* Autoclose resources with a try-with-resources statement
* Recognizing common exception classes and categories
* Creating custom exceptions

**Day 4**

**Lambda Expressions**

* Behavior Parametrization
* Anonymous inner classes
* Defining a Lambda Expression

**Lambda Built-in Functional Interfaces**

* Core interfaces - Predicate, Consumer, Function, Supplier
* Using primitive versions of base interfaces
* Using binary versions of base interfaces

**Collections Streams, and Filters**

* Describing the Stream interface
* Iterating through a collection using lambda syntax
* Filtering a collection using lambda expressions
* Calling an existing method using a method reference
* Chaining multiple methods together
* Defining pipelines in terms of lambdas and collections

**Lambda Operations**

* Extracting data from an object using map
* Describing the types of stream operations
* Describing the Optional Interface
* Sorting a stream
* Saving results to a collection using the collect method

**Day 5**

**Concurrency**

* Describing operating system task scheduling
* Creating worker threads using Runnable and Callable
* Using an ExecutorService to concurrently execute tasks
* Identifying potential threading problems
* Using synchronized and concurrent atomic to manage atomicity

**Spring Introduction**

* Shortcomings of Java EE and the Need for Loose Coupling
* Managing Beans, The Spring Container, Inversion of Control
* The Factory Pattern
* Configuration Metadata
* Annotation Configuration @Autowired, @Required
* @Component
* @Value and @Qualifier
* Java Configuration, @Configuration, XML free configuration
* The AnnotationConfigApplicationContext

**Day 6**

**Introduction to REST**

* The GET,POST,PUT & DELETE methods
* URI Templates
* Rest and Spring MVC
* Restful Services using Spring
* @RequestMapping, @PathVariable, @RequestBody
* Controllers with @RestController