



Course Goals and Non Goals



➤ Course Goals

- Implementing OOPs features in Java
- Developing Java Desktop Applications
- Use of Core JDK 1.8 API
- Testing using Junit 4
- Implementing Multithreading

➤ Course Non Goals

- Developing GUI applications



Pre-requisites



Basic Programming Concepts
OOPs
XML

Intended Audience



Developers new to Java technology



Day Wise Schedule



Day 1

- Lesson 1: Introduction to Object-Oriented technology
- Lesson 2: Objects and Classes
- Lesson 3: Principles in Object-Oriented technology
- Lesson 4: Some more concepts in OOP

Day 2

- Lesson 5: Introduction to Java
- Lesson 6: Eclipse 4.4 (Luna) as an IDE
- Lesson 7: Language Fundamentals

Day 3

- Lesson 8: Classes and Objects
- Lesson 9: Exploring Basic Java Class Libraries

Day 4

- Lesson 10: Inheritance and Polymorphism
- Lesson 11: Abstract Classes and Interfaces

Day Wise Schedule



Day 5

- OCA Preparation on Basic Concepts

Day 6

- Lesson 12 : Exception Handling

Day 7

- Lesson 13: Array
- Lesson 14: Collection

Day 8

- Lesson 14:Collection Continue..

Day 9

- Lesson 14:Collection Continue..
- Lesson 15 : Generics

Day Wise Schedule



Day 10

- OCA Preparation on Collection and Practice

Day 11

- Lesson 16: Multithreading
- Lesson 17: File IO

Day 12

- Lesson 18: Property Files
- Lesson 19: Introduction to Junit

Day 13

- Lesson 20: Lambda Expression
- Lesson 21: Stream API

Day Wise Schedule



Day 14

- OCA Preparation on Collection and Practice

Day 15

- OCA Preparation

Table of Contents



Lesson 1: Introduction to Object-Oriented Technology

- 1.1: Object Oriented concepts
 - 1.1.1: What is Object-Oriented Programming?
 - 1.1.2: Why Object-Oriented Programming?

Lesson 2: Objects and Classes

- 2.1: What is an Object?
 - 2.1.1(Object State, Object Behavior, Object Identity)
- 2.2: What is a Class?
 - 2.2.1: Getting into Details
 - (Class Attribute and Operations, Access Modifiers, Constructors and Destructors, Attribute Types)

Table of Contents



Lesson 3: Principles in Object-Oriented Technology

- 3.1: Object-Oriented Principles
 - 3.1.1: Abstraction
 - 3.1.2: Encapsulation
 - 3.1.3: Modularity
 - 3.1.4: Hierarchy
- 3.2: Polymorphism

Lesson 4: Some More Concepts in OOP

- 4.1: Static Members
- 4.2: Abstract Class
- 4.3: Interface
- 4.4: Packages

Table of Contents



Lesson 5: Introduction to Java

- 5.1: Introduction to Java
- 5.2: Features of Java
- 5.3: Simple Program in Java
- 5.4: Developing software in Java

Lesson 6: Eclipse 4.4 (Luna) as an IDE

- 6.1: Installation and Setting up Eclipse
- 6.2: Introduction to Eclipse IDE
- 6.3: Creating and Managing Java Projects
- 6.4: Miscellaneous Options

Table of Contents



Lesson 7: Language Fundamentals

- 7.1: Keywords
- 7.2: Primitive Data Types
- 7.3: Operators and Assignments
- 7.4: Variables and Literals
- 7.5: Flow Control: Java's Control Statements
- 7.6: Best Practices

Lesson 8: Classes and Objects

- 8.1: Classes and Objects
- 8.2: Packages
- 8.3: Access Specifiers
- 8.4: Constructors - Default and Parameterized
- 8.5: this reference
- 8.6: Memory management in java
- 8.7: using static keyword
- 8.8: Enum
- 8.9: Best Practices

Table of Contents



Lesson 9: Exploring Basic Java Class Libraries

- 9.1: The Object Class
- 9.2: Wrapper Classes
- 9.3: Type casting
- 9.4: Using Scanner Class
- 9.5: System Class
- 9.6: String Handling
- 9.7: Date and Time API
- 9.8: Best Practices

Lesson 10: Inheritance and Polymorphism

- 10.1: Inheritance
- 10.2: Using super keyword
- 10.3: InstanceOf Operator
- 10.4: Method & Constructor overloading
- 10.5: Method overriding
- 10.6: @override annotation
- 10.7: Using final keyword

Table of Contents



Lesson 11: Abstract Classes and Interfaces

- 11.1: Abstract class
- 11.2: Interfaces
- 11.3: default methods
- 11.4: static methods on Interface
- 11.5 : Interface rules
- 11.6: Abstract class Vs Interface
- 11.7 :Anonymous Classes
- 11.8: Runtime Polymorphism

Table of Contents



Lesson 12: Exception Handling

- 12.1: Introduction
- 12.2: Exception Types and Exception Hierarchy
- 12.3: Try-catch-finally
- 12.4: Try-with-resources
- 12.5: Multi catch blocks
- 12.6: Throwing exceptions using throw
- 12.7: Declaring exceptions using throws
- 12.8: User defined Exceptions
- 12.9: Best Practices

Table of Contents



Lesson 13: Array

- 13.1: One dimensional array
- 13.2: Multidimensional array
- 13.3: Using varargs
- 13.4: Using Arrays class
- 13.5: Best Practices

Lesson 14: Collection

- 14.1: Collections Framework
- 14.2: Collection Interfaces
- 14.3: Iterating Collections
- 14.4: Implementing Classes
- 14.5: Comparable and Comparator
- 14.6: Hashtable , HashMap, TreeMap
- 14.7: Best Practices

Table of Contents



Lesson 15: Generics

- 15.1: Generics
- 15.2: Writing Generic Classes
- 15.3: Using Generics with Collections
- 15.4: Best Practices

Lesson 16 : Multithreading

- 16.1 Understanding Threads
- 16.2 Thread life cycle
- 16.3 Scheduling threads- Priorities
- 16.4 Controlling threads using sleep(),join()
- 16.5 :Consumer Producer Problem
- 16.6 :Inter Thread Communication : wait ,notify,notifyAll methods
- 16.7 : Synchronization Problem
- 16.8 :Thread Interaction

Table of Contents



Lesson 17 : File IO

- 17.1: Overview of I/O Streams
- 17.2: Types of Streams
- 17.3: The Byte-stream I/O hierarchy
- 17.4: Character Stream Hierarchy
- 17.5: Buffered Stream
- 17.6: The File class
- 17.7: The Path class
- 17.8: Object Stream
- 17.9: Best Practices

Lesson 18 : Property Files

- 18.1: What are Property Files?
- 18.2: Types of Property files
- 18.3: User defined Properties

Table of Contents



Lesson 19 : Introduction to JUnit 4

- 19.1: Introduction
- 19.2: JUnit
- 19.3: Installing and Running JUnit
- 19.4: Testing with JUnit
- 19.5: Testing Exceptions
- 19.6: Test Fixtures
- 19.7: Advanced Testing Concepts
- 19.8: Test Suites
- 19.9 :Parameterized Tests
- 19.10: Mocking Concept using EasyMock Framework

Lesson 20 : Lambda Expression

- 20.1: Introduction to Functional Interface
- 20.2: Writing Lambda Expressions
- 20.3: Built in Functional Interfaces
- 20.4: Built in Functional Interfaces and Lambda Expressions
- 20.5: Method reference

Table of Contents



Lesson 21: Stream API

- 21.1: Introduction to Stream API
- 21.2: Working with Stream API
- 21.3: Stream Operations

References



Books:

- Java, The Complete Reference; by Herbert Schildt
- Thinking in Java; by Bruce Eckel
- Beginning Java 8 Fundamentals by Kishori Sharan

Websites:

- Java home page: <http://java.sun.com/>
- JDK 1.8 documentation: <http://docs.oracle.com/javase/8/docs>
- Multithreading :
<https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html>



Next Step Courses



Servlets
JSP



Other Parallel Technology Areas



C ++
C#.Net
Visual Basic.Net