



Course Brochure

Core Java

Overview

•Core Java training course is intended for students without an extensive programming background. It covers language fundamentals and implementation of pure object oriented programming principles using java API and developing GUI applications. This course also covers how to use built in java classes and how to apply them in the real time environment.

Pre-requisites

•Basic programming knowledge in any languages like c or c++ is must.

Applications

COURSE CONTENTS

Introduction

- ❖ Programming language Types and Paradigms.
- ❖ Why Java?
- ❖ Flavors of Java.
- ❖ Java Designing Goal.
- ❖ Role of Java Programmer in Industry.
- ❖ Features of Java Language.
- ❖ JVM The heart of JAVA.

Language Fundamentals

- ❖ **The Java Environment.**
 - Installation
 - Java Program Development
 - Java Source File Structure
 - Compilation
 - Executions
- ❖ **Basic Language Elements.**
 - Lexical Tokens, Identifiers
 - Keywords, Literals, Comments
 - Primitive Data types
 - Operators
 - Condition Statements
 - Control Statements
 - Arrays
 - Command line Arguments
 - OCJP

Object Oriented Programming

- ❖ OOPS Fundamentals.
- ❖ Object & Object reference.
- ❖ Constructors.
- ❖ Method Overloading, Recursion.
- ❖ Access Specifiers & Access Modifiers.
- ❖ Design of Accessor and Mutator Methods.
- ❖ Inheritance
- ❖ Inner Class & Anonymous Classes
- ❖ Abstract Class
- ❖ Interfaces
- ❖ Cloning Objects, shallow and deep cloning

Packages

- ❖ Organizing Classes and Interfaces in Packages
- ❖ Package as Access Protection
- ❖ Defining Packages
- ❖ CLASSPATH Setting for Packages
- ❖ Making JAR Files for Library Packages
- ❖ Import and Static Import
- ❖ Naming convention for Packages

Exception Handling

- ❖ The idea behind Exception
- ❖ Exceptions & Errors
- ❖ Types of Exception
- ❖ Control Flow In Exceptions
- ❖ JVM reaction to Exceptions
- ❖ Use if try, catch, finally, throw, throws in Exception Handling
- ❖ In-built and User Defined Exceptions
- ❖ Checked and Un-Checked Exceptions

Multi Threading

- ❖ Understanding Threads
- ❖ Needs of Multi-threaded Programming
- ❖ Thread Life-Cycle
- ❖ Creating Child Threads
- ❖ Multi Threads in a program
- ❖ Thread Priorities
- ❖ Synchronizing Threads
- ❖ Inner Communication of Threads
- ❖ Critical Factor in Thread DeadLock

Input/Output Operation in Java(java.io Package)

- ❖ Streams and the new I/O Capabilities
- ❖ Understanding Streams
- ❖ The Classes for Input and Output
- ❖ The Standard Streams
- ❖ Working with File Object
- ❖ File I/O Basics
- ❖ Reading and Writing to Files
- ❖ Buffer and Buffer Management
- ❖ Read/Write Operations with File Channel
- ❖ Formatted Input/Output
- ❖ Sequence Input
- ❖ Random Access
- ❖ Serializing Objects
- ❖ Character Streams

GUI Programming

- ❖ Designing Graphical User Interfaces in Java
- ❖ Components and Containers
- ❖ Basics of Components
- ❖ Using Containers
- ❖ Layout Managers
- ❖ AWT Components
- ❖ Event-Driven Programming in Java
- ❖ Adapter Classes as Helper Classes in Event Handling
- ❖ Anonymous Inner classes a Short cut to Event Handling
- ❖ Adding A Menu to Window
- ❖ Dialog Boxes
- ❖ Built-in Dialog Boxes – FileDialog
- ❖ Extending GUI Features Using Swing Components

APPLETS

- ❖ Applet & Application
- ❖ Applet Architecture
- ❖ Parameters to Applet
- ❖ Embedding Applets in Web page
- ❖ Applet Security Policies

Collections Framework & Utility Classes

- ❖ Utility Methods for Arrays
- ❖ String Tokenizer
- ❖ Observable and Observer Objects
- ❖ Date & Time
- ❖ Data structures
- ❖ Timer and Timer Task for Job Scheduling
- ❖ Using Scanner
- ❖ Regular Expression
- ❖ Collections of Objects
- ❖ Collection Types
- ❖ List
- ❖ Set
- ❖ Sequence
- ❖ Map
- ❖ Understanding Hashing
- ❖ Use of ArrayList & Vector
- ❖ Hashtable & HashMap

Annotations:

Generics:

Reflection:

Java SE 6.x/ 7.x Features