

Java 2 Enterprises Edition

Duration: 90 hrs

What we will learn:

The Java programming language course provides students with a solid foundation of Java programming with Java, including information about the syntax of Java programming language; object oriented with Java programming language creating graphical user interface (GUIs), exceptions file input/output (I/O), the thread and networking. Programmer familiar with object – oriented concept can learn how to develop Java technology applications. The course features the Java platform, Standard Edition 6 (Java SE 6) platform, and utilized the Java SE development Kit 6 (JDK 6) product. The student performs the course lab experience using MyEclipse Integrated development Environment (IDE).

Learn To:

Install JAVA IDE

Basic coding standards

Database connectivity

Application Development

Web Development

Deploying software/Application

Scope:

Software Architect

Windows Application Developer

Web Application Developer

Java Programmer

Technical consultant

Software Associate

Software Engineer

Prerequisites

Basic Programming Knowledge

Course Objectives

To understand;

Basic functioning in Java

Web & window application development

Database connectivity development

Deploying the Java developed application.

Course Topics:

Introduction and overview of JAVA

Getting started

- Introduction to Java technology
- Getting to know your Java programming environment
- Java programming basics
- Complete Programming Syntax

Learning the basics of Java

- Getting input from keyboard
- Control Structure
- Java Arrays
- Command-line arguments
- Working with built-in Java Classes

Classes and Objects

- Creating your own Classes
- Packages and Java class path
- JavaBeans

Object-Oriented Programming with Java

- The object paradigm
- Object-oriented (OO) programming
- Encapsulation, inheritance and polymorphism
- OO analysis and design: "Is a" and "Has a" relationships
- Designing an OO application step by step
- Diagramming object structure with Unified Modeling Language (UML)

Basic features of Java

- Exception Handling
- Multi Threading

New features of Java Language

- Generics and collection framework
- Enhanced for Loop
- Auto boxing/Unboxing
- Static Import
- Metadata (Annotations)

IO package:

The Java language provides a simple model for input and output (I/O). All I/O is performed by writing to and reading from streams of data. The data may exist in a file or an array, be piped from another stream, or even come from a port on another computer. The flexibility of this model makes it a powerful abstraction of any required input and output.

Storing and Retrieving Data with I/O

Java streams

- Streams, Readers and Writers
- Accessing files
- Formatting text output

Files and directories

- Reading and writing files
- Creating, deleting and renaming files
- Obtaining directory and file information

SQL Package(JDBC)

The Java Database Connectivity (JDBC) specification is a new basis for developers to interface with data sources. The structure of the JDBC is designed to allow developers to program for a standard interface, and let the low-level JDBC driver for different databases deal with the connection and querying of the databases; hence, the developer does not need to worry about dealing with different SQL flavors for each database. The JDBC is also very flexible—a developer doesn't necessarily lose features specific to the target database. Best of all, the JDBC is based on the Java language!

JDBC database access

- Leveraging the JDBC API
- Choosing database drivers
- Connecting to a database

Improving performance with prepared statements and stored procedures

Submitting SQL statements

Retrieving and processing results

GUI programming

There are two basic types of GUI program in Java: stand-alone applications and applets. An applet is a program that runs in a rectangular area on a Web page. Applets are generally small programs, meant to do fairly simple things, although there is nothing to stop them from being very complex. Applets were responsible for a lot of the initial excitement about Java when it was introduced, since they could do things that could not otherwise be done on Web pages. However, there are now easier ways to do many of the more basic things that can be done with applets, and they are no longer the main focus of interest in Java

- AWT and Event Handling
- Swing Overview

Developing Java Web Applications

1. Servlet

Servlets were Java's first server side web technology. It was meant as an alternative to the early web technology "CGI scripts". CGI Scripts were programs that were executed on the web server when a request arrived which was to be handled by that CGI script. Thus, a CGI script process was to be started and finished whenever a request arrived at the web server for a CGI script. Starting a new process is rather slow, and can be memory intensive. Therefore Sun came up with Sevlets. A CGI script could be written in many languages

- Introduction and Overview
- Web application development
- Key components of Java EE for Web development
- Utilizing the Java Servlet API
- Creating Web-based forms for user input
- Extending servlets with Java Database Connectivity (JDBC)
- Maintaining State in Java Web Applications
- Reading and writing cookies to personalize Web content

2. JSP

JSP is short for Java Server Pages. JSP was a reaction to the complaints that embedding HTML inside Servlets (Java code) was a bad idea. If the layout across a site needed to be changed, you would have to do so in the Java code. This wasn't always that easy to do, since HTML generating code was interleaved with domain logic code. JSP was also a reaction to the then new web technologies ASP (Active Server Pages) from Microsoft

- Creating Java Server Pages (JSP) JSP fundamentals
- State management with JSP
- Designing JSP and Servlet Architectures
- Integrating JSP and JavaBeans
- Calling JSP from servlets
- Integrating JSP Custom Tags
- Leveraging the Tag Extension mechanism
- Developing custom tags

EJB 3.0:

EJB has seen a dramatic improvement in usability with the 3.0 release that's part of Java EE 5, and it looks as if this might bring a resurgence of interest. One thing to understand about this course is that it is geared primarily to the student with no prior EJB experience and not so much to the EJB 2.x developer looking for a skills upgrade, although that is certainly a possible class of participant. A bit more on this in the Timeline section, but basically we're presenting EJB 3.0, fresh and unencumbered by its past, to experienced Java developers – preferably Java 5 since we'll use annotations so heavily, but Java-1.4 programmers shouldn't have any big trouble.

- Implementing the Business Tier with EJB 3.0
- Stateless and Stateful beans
- Applying advanced strategies of session beans
- Obtaining asynchronous communication with JMS
- Unleashing Message Driven Beans (MDB)

Java 2 Enterprises Edition (Advanced with Spring and Hibernate Framework)

Duration: 35 hrs.

Struts 2.0

Apache Struts 2 is an elegant, extensible framework for creating enterprise-ready Java web applications. The framework is designed to streamline the full development cycle, from building, to deploying, to maintaining applications over time.

Apache Struts 2 was originally known as Web Work 2. After working independently for several years, the Web Work and Struts communities joined forces to create Struts². This new version of Struts is simpler to use and closer to how Struts was always meant to be.

MVC Framework

- Struts 2 Basics
- Struts 2 and Ajax
- Struts 2 and Tiles
- Struts 2 and view technologies
- Struts 2 Tag Libraries

Developing Enterprise Java Applications with Spring 2.5 and Hibernate 3.5

Spring 2.5:

Spring Framework is a Java platform that provides comprehensive infrastructure support for developing Java applications. Spring handles the infrastructure so you can focus on your application.

- Introducing the Spring Framework
- Spring architecture fundamentals
- Applying Inversion of Control (IOC) and Dependency Injection (DI)
- Minimizing Code with Aspect-Oriented Programming (AOP)
- Transparently applying common functionality
- Constructing an Effective Data Access Tier with Spring
- Simplifying data access with JDBC templates
- Abstracting the Data Access Layer
- Building a Web Tier with Spring MVC
- Processing Web requests

Hibernate 3.5:

Hibernate is an open source object/relational mapping tool for Java. Hibernate lets you develop persistent classes following common Java idiom - including association, inheritance, polymorphism, composition and the Java collections framework

- Persisting Objects with Hibernate
- Integrating Hibernate
- Generating Hibernate applications
- Handling Complex Object Relationships
- The role of the Hibernate Session
- Mapping collections
- Strategies for building object associations
- Effectively mapping inheritance relationships
- Optimizing Data Access
- Applying Hibernate Query Language (HQL)

Web Services:

Ajax

Ajax Basics & Development Tools

JavaScript Basics and DOM APIs

JavaScript Advanced

Web Services

Web services overview

Web Services Overview

XML

XML Namespace

X2ML Schema

XML Schema Design Pattern

XPath

XSLT

XSLT 2

Java APIs for XML parsing and transformation

JAXP, SAX, DOM

StAX

SOAP and WSDL

SOAP

SOAP Processing Model

SOAP Protocol Binding

WSDL Basics

WSDL Bindings

SOAP UI Basics

SAAJ

Server

Apache Tomcat Web Server

BEA Web logic Application Server