

Dear Student,

Based upon your enquiry we are pleased to send you the course curriculum for Java

Given below is the brief description for the course you are looking for:

Core Java

❖ Fundamentals of Object-Oriented Programming

- ✓ Procedure-oriented Vs Object oriented Programming.
- ✓ Basic concepts of Object-Oriented Programming -
 - Object and classes.
 - Data Abstraction and Encapsulation
 - Inheritance
 - Polymorphism
 - Dynamic Binding.
- ✓ Benefits of OOP
- ✓ Applications of OOP

❖ JAVA Evolution

- ✓ Java History
- ✓ Why Java? Java Features

❖ Constants, Variables, and Data Types.

❖ Operators and Expressions.

❖ Decision Making and Branching.

❖ Decision Making and Looping.

❖ Classes, Objects and Methods.

- ✓ Fields and Method Declaration.
- ✓ Creating Objects.
- ✓ Accessing Class Members.
- ✓ Constructors and Method Overloading
- ✓ Inheritance
- ✓ Overriding Methods
- ✓ Final Variables and Methods.
- ✓ Final Classes
- ✓ Finalizer Methods
- ✓ Abstract Methods and classes.
- ✓ Visibility Control

❖ Arrays, Strings and vectors

❖ Interfaces: Multiple Inheritance

- ✓ Defining Interfaces
- ✓ Extending Interfaces
- ✓ Implementing Interfaces
- ✓ Accessing Interface Variables

❖ Packages-Putting Class Together

- ✓ Java API Packages
- ✓ Using System Packages
- ✓ Creating and Accessing a Package
- ✓ Adding a class to a package
- ✓ Hiding Classes

❖ Managing errors and Exceptions

- ✓ Types of Errors- Compile-Time and Run –Time
- ✓ Syntax of Exception Handling Code
- ✓ Multiple catch Statements
- ✓ Using finally Statement
- ✓ Throwing Our Own exceptions

❖ File Handling

❖ MultiThreaded Programming

- ✓ What are the threads?
- ✓ Creating Threads
- ✓ Extending the Thread Class
- ✓ Stopping and Blocking a Thread
- ✓ life Cycle of a Thread
- ✓ Using Thread Methods
- ✓ Thread Exceptions
- ✓ Thread Priority
- ✓ Implementing the 'Runnable' Interface

❖ Applet programming

❖ Java Collection

- ✓ Introduction
- ✓ The Collection Interface
- ✓ The Set Interface and Classes in Details
- ✓ The List Interface and Classes in Details

Advanced Java

❖ Generics

- ✓ Introduction to Generics
- ✓ Type Erasure
- ✓ Type Boundaries
- ✓ Wildcards
- ✓ Generic Methods
- ✓ Strengths and Weaknesses of Generics
- ✓ Legacy Code and Generics

❖ Threads

- ✓ Java Thread Model
- ✓ Creating and Running Threads
- ✓ Manipulating Thread State
- ✓ Thread Synchronization
- ✓ Volatile Fields vs. Synchronized Methods
- ✓ wait and notify
- ✓ join and sleep
- ✓ The Concurrency API
- ✓ Atomic Operations

❖ Reflection

- ✓ Uses for Meta-Data
- ✓ The Reflection API
- ✓ The Class<T> Class
- ✓ The java.lang.reflect Package
- ✓ Reading Type Information
- ✓ Navigating Inheritance Trees
- ✓ Dynamic Instantiation
- ✓ Dynamic Invocation

- ✓ Reflecting on Generics

❖ Annotations

- ✓ Aspect-Oriented Programming and Java
- ✓ The Annotations Model
- ✓ Annotation Types and Annotations
- ✓ Built-In Annotations
- ✓ Annotations vs. Descriptors (XML)

❖ Sockets

- ✓ The OSI Reference Model
- ✓ Network Protocols
- ✓ The Socket Class
- ✓ The ServerSocket Class
- ✓ Connecting Through URL Objects
- ✓ HTTP and Other TCP Servers
- ✓ Datagram Clients and Servers
- ✓ Non-Blocking Sockets

❖ Database and SQL Fundamentals

- ✓ Relational Databases and SQL
- ✓ Database, Schema, Tables, Columns and Rows
- ✓ SQL Versions and Vendor Implementations
- ✓ DDL -- Creating and Managing Database Objects
- ✓ DML -- Retrieving and Managing Data
- ✓ Sequences
- ✓ Stored Procedures
- ✓ Using SQL Terminals

❖ JDBC Fundamentals

- ✓ What is the JDBC API?
- ✓ JDBC Drivers
- ✓ Making a Connection
- ✓ Creating and Executing a Statement
- ✓ Retrieving Values from a ResultSet
- ✓ SQL and Java Datatypes
- ✓ Creating and Updating Tables
- ✓ Handling SQL Exceptions and Proper Cleanup
- ✓ Handling SQLWarning

❖ Advanced JDBC

- ✓ SQL Escape Syntax
- ✓ Using Prepared Statements
- ✓ Using Callable Statements
- ✓ Scrollable Result Sets
- ✓ Updatable Result Sets
- ✓ Transactions
- ✓ Commits, Rollbacks, and Savepoints
- ✓ Batch Processing

❖ Introduction to Row Sets

- ✓ Row Sets in GUI and J2EE programming
- ✓ Advantages of RowSets
- ✓ RowSet Specializations
- ✓ Using CachedRowSets

❖ Design Patterns

- ✓ What are Design Patterns?
- ✓ Singleton, Factory Method, Abstract Factory



-
- ✓ Adapter, Composite, Decorator
 - ✓ Chain of Responsibility, Observer / Publish-Subscribe, Strategy, Template
 - ✓ Data Access Object (DAO)

Tech Altum

JSP and Servlets

❖ Overview of Servlets and JSP

- ✓ What is a Java servlet?
- ✓ What advantage does a Java servlet have over a traditional CGI program?
- ✓ What is JSP?
- ✓ What are the advantages of JSP?

❖ Installation and Setup

❖ Servlets

- ✓ Basic Servlet Structure
- ✓ A Simple Servlet Generating Plain Text
- ✓ Compiling and Invoking the Servlet
- ✓ A Simple Servlet Generating HTML
- ✓ Some Simple HTML Utilities for Servlets

❖ Processing the Request: Form Data

- ✓ Introduction (Format, URL-encoding, GET, POST)
- ✓ Example: Reading Specific Parameters
- ✓ Example: Making Table of All Parameters

❖ Processing the Request: HTTP Request Headers

- ✓ Common Request Headers and Their Meaning
- ✓ Reading Headers from Servlets
- ✓ Example: Making Table of All Headers

❖ Accessing the Standard CGI Variables

- ✓ The CGI Variables, Their Meaning, and the Servlet Equivalent
- ✓ Example: Making a Table of All CGI Variables

❖ Generating the Response: HTTP Status Codes

- ✓ Overview: Status Codes and Messages
- ✓ Setting Status Codes from Servlets
- ✓ HTTP 1.1 Status Codes and Their Meanings
- ✓ Example: A Search Engine Front End

❖ Generating the Response: HTTP Response Headers

- ✓ Overview
- ✓ Common Response Headers and Their Meaning
- ✓ Example: Starting a Long-Running Calculation, Showing Partial Results, and Periodically Updating Display with New Data.

❖ Handling Cookies

- ✓ Intro: How Cookies Can Be Used (and Abused)
- ✓ The Servlet Cookie API
- ✓ Some Cookie Utilities
- ✓ Example: A Customized Search Engine Interface

❖ Session Tracking

- ✓ Overview of Session Tracking
- ✓ The Servlet Session Tracking API
- ✓ Example

❖ JavaServer Pages (JSP)

- ✓ JSP Overview
- ✓ JSP Syntax Summary
- ✓ Template Text (Static HTML)
- ✓ JSP Scripting Elements: Expressions, Scriptlets, and Declarations
- ✓ JSP Directives
- ✓ Example using JSP Scripting Elements and Directives
- ✓ Predefined Variables
- ✓ JSP Actions
- ✓ JSP Comments and Character Escaping Conventions

- ✓ Creating Custom JSP Tag Libraries

J2EE Struts 2

- ❖ **Struts 2 Features**
- ❖ **Struts 2 History**
- ❖ **Struts 2 Architecture**
- ❖ **Why Struts 2**
- ❖ **Struts 1.x Vs Struts 2.x**
- ❖ **Downloading and Installing Struts 2**
- ❖ **Struts 2 Hello World Application**
- ❖ **Creating the development directory Structure**
- ❖ **Writing Jsp, Java and Configuration files.**
- ❖ **Struts 2 xml configuration file**
- ❖ **Struts 2 Actions**
- ❖ **Struts 2 Actions Example.**
- ❖ **Struts 2 Actions Introduction**

❖ **Struts 2 Redirect Action**

Struts 2 Login Application

❖ **Developing Login Application in Struts 2**

❖ **Adding Validation to Struts 2 Login Application**

❖ **Running and testing application**

❖ **Client Side Validation in Struts 2 Login Application**

❖ **Validations using Struts 2 Annotations**

❖ **Login/Logout With Session**

❖ **Connecting to MySQL Database in Struts 2 (Struts 2 MySQL).**

❖ **Validate TextArea**

Struts 2 Tags

Spring

❖ Spring 3 .0

- ✓ Introduction
- ✓ Spring 3.0 Features
- ✓ Spring 3 Hello World Example
- ✓ @configuration annotation in Spring 3

❖ Spring 2.5

- ✓ Introduction to the Spring Framework
- ✓ Downloading and Installing Spring
- ✓ Installing Spring Framework
- ✓ Spring Injection Example
- ✓ Spring Batch Example
- ✓ Inheritance in Spring
- ✓ Spring Hello World Application
- ✓ Calling Constructor in Spring.
- ✓ init Method in Spring
- ✓ Bean life cycle in spring

❖ Spring 2.5 MVC

- ✓ Spring MVC Introduction
- ✓ How Spring MVC Works
- ✓ Spring MVC Controllers hierarchy
- ✓ Spring MVC Getting Started
- ✓ Spring MVC Hello World
- ✓ Spring 2.5 MVC User Registration Example
- ✓ AbstractController in Spring MVC
- ✓ Controller Interface
- ✓ Parameterizable View Controller
- ✓ MultiActioncontroller Example
- ✓ UriFileNameviewController
- ✓ AbstractWizardFormController
- ✓ SimpleFormController
- ✓ InternalResourceViewResolver example
- ✓ ResourceBundleViewResolver example
- ✓ XmlViewResolver Example
- ✓ Chaining ViewResolvers example
- ✓ UriBasedViewResolver Example
- ✓ BeanNameUrlHandlerMapping
- ✓ BeanNameUrlHandlerMappingWithCommandClass
Example
- ✓ Spring MVC Tags

Hibernate 3.0

- ❖ Introduction to Hibernate 3.0
- ❖ Hibernate Architecture
- ❖ First Hibernate Application
- ❖ Running the Example in Eclipse.
- ❖ Understanding Hibernate O/R Mapping
- ❖ Understanding Hibernate <generator> element
- ❖ Using Hibernate <generator> to generate id incrementally
- ❖ Hibernate Update Query
- ❖ Hibernate Delete Query

Hibernate Query Language

- ❖ Introduction to Hibernate Query Language
- ❖ Preparing table for HQL Examples
- ❖ Developing POJO class
- ❖ HQL from Clause
- ❖ HQL Select Clause
- ❖ Hibernate Count Query
- ❖ Hibernate Avg() Function (Aggregate Functions)
- ❖ Hibernate Max() Function (Aggregate Functions)
- ❖ Hibernate Min() Function (Aggregate Functions)
- ❖ HQL Where Clause
- ❖ HQL Group By Clause
- ❖ HQL Order By Example

Hibernate Criteria Query

- ❖ Hibernate Criteria Query Example
- ❖ Narrowing the result set
- ❖ Hibernate's Built-in criterion: Between (using Integer)
- ❖ Hibernate's Built-in criterion: Between (using Date)

- ❖ Hibernate Criteria Expression (eq)
- ❖ Hibernate Criteria Expression (lt)
- ❖ Hibernate Criteria Expression (le)
- ❖ Hibernate Criteria Expression (gt)
- ❖ Hibernate Criteria Expression (ge)
- ❖ Hibernate Criteria Expression (and)
- ❖ Hibernate Criteria Expression (or)
- ❖ Insert Data into Database Using Hibernate Native SQL
- ❖ Hibernate Native SQL Example
- ❖ Associations and Joins
- ❖ Hibernate Aggregate Functions (Associations and Joins)
- ❖ Hibernate Subqueries

Hibernate Projections

- ❖ Hibernate Projections
- ❖ Hibernate Projections (rowCount or countDistinct)
- ❖ Hibernate Projection Count
- ❖ Hibernate Projection Example (Sum).
- ❖ Hibernate How To?

- ❖ Hibernate types