GTyde Technologies

Your Success Is Our Success

Syllabus of Core Java Course

Module 1: Introduction to Java

- Features of Java
- Simple
- Secure
- Portable
- Robust
- Multithreading
- Platform-Independent
- Distributed.
- Dynamic
- New Features of Java 8
- Introducing Java Environment
- Java Development Kit
- Java Platforms
- Java Virtual Machine
- Java API
- Java Programs
- Installing Java
- What about CLASSPATH?
- Java's Reserve Words
- Starting a Java program
- Line 2—public static void main(String[] args)
- Line 3—System.out.println("Hello from Java!");
- Compiling Code 15
- Compiling Code: Using Command-Line Options
- Cross-Compilation Options
- Compiling Code: Checking for Deprecated Methods
- Running Code

- Running Code: Using Command-Line Options
- Commenting Your Code
- Importing Java Packages and Classes
- Finding Java Class with CLASSPATH
- Summary

Module 2: Variables, Arrays and Strings

- Variables
- Data Typing
- Arrays
- Strings
- What Data Types are Available?
- Creating Integer Literals
- Creating Floating-Point Literals
- Creating Boolean Literals
- Creating Character Literals
- Creating String Literals
- Creating Binary Literals
- Using Underscores in Numeric Literals
- Declaring Integer Variables
- Declaring Floating-Point Variables
- Declaring Character Variables
- Declaring Boolean Variables
- Initializing Variables Dynamically
- Conversion between Data Types
- Automatic Conversions
- Casting to New Data Types
- Declaring One-Dimensional
- Creating One-Dimensional Arrays
- Initializing One-Dimensional Arrays
- Declaring Multi-Dimensional Arrays
- Creating Multi-Dimensional Arrays
- Initializing Multi-Dimensional Arrays
- Creating Irregular Multi-Dimensional Arrays
- Getting an the Length of an Array
- Understanding General Form of Static Import

- Importing Static Members
- The String Class
- Getting String Length
- Concatenating Strings
- Getting Characters and Substrings
- Searching For and Replacing Strings
- Changing Case in Strings
- Checking for Empty String
- Formatting Numbers in Strings
- The StringBuffer Class
- Creating StringBuffers
- Getting and Setting StringBuffer Lengths and Capacities
- Setting Characters in String Buffers
- Appending and Inserting Using StringBuffers
- Deleting Text in StringBuffers
- Replacing Text in String Buffer
- Using the Wrapper Class
- Autoboxing and Unboxing of Primitive Types
- Learning the Fundamentals of Varargs Methods
- Overloading Varargs Methods
- Learning the Ambiguity in Varargs Methods
- Using Non-Reifiable Formal Parameters

Module 3: Operators, Conditionals and Loops

- Operators
- Conditionals
- Loops
- Operator Precedence
- Incrementing and Decrementing (++ and --)
- Unary NOT (~ And !)
- Multiplication and Division (* and /)
- Modulus (%)
- Addition and Subtraction (+ and -)
- Shift Operators (>>, >>>, and <<)
- Relational Operators (>, >=, <, <=, ==, and !=)
- Bitwise and Bitwise Logical AND, XOR, and OR (&, ^, and /)

- Logical (&& and ||)
- The if-then-else Operator
- Assignment Operators (= and [operator]=)
- Using the Math
- Changes in the Math Class
- Class StrictMath
- Comparing Strings
- The if Statement
- The else Statement
- Nested if
- The if-else Ladders
- The switch Statement
- Using Strings in switch Statement
- The while Loop
- The do-while Loop
- The for Loop
- The for-each Loop
- Supporting for-each in Your Own Class
- A (Poor) Solution
- Significance of for-
- Nested Loops
- Using the break Statement
- Using the continue Statement
- Using the return Statement
- Summary

Module 4: Class, Object, Packages and Access Specifiers

- The Control Overview of a Class
- Working with Objects
- Working with Methods
- Defining Default Methods
- Working with Constructors
- Using Default Constructor
- Using Parameterized Constructors
- Exploring Packages
- Studying the Types of Packages

- Importing Packages
- Using Access Specifiers
- Working with Streams API
- Stream API Overview
- Collection and Stream
- Commonly Used Functional Interfaces in Stream
- java.util.Optional
- Aggregate Operations
- Working with Time API

Module 5: Implementing Object-Oriented Programming in Java

- Understanding Encapsulation
- Understanding Abstraction
- Understanding Inheritance
- Understanding the final Keyword
- Preventing Inheritance
- Declaring Constant
- Preventing Method Overriding
- Implementing Interfaces
- Working with Lambda Expressions
- Method References
- Using Lambda Expressions
- Implementing Abstract Classes and Methods
- Difference between Abstract Classes and Interfaces
- Implementing Polymorphism
- Understanding the Static Polymorphism
- Understanding the Dynamic Polymorphism
- Summary

Module 6: Working with Streams, Files and I/O Handling

- Streams, Readers and Writers
- Essentials in NIO
- Buffers
- Channels
- Charsets and Selectors
- Enhancements in NIO with Java 8

- The Path Interface
- The Files Class
- The Paths Class
- The File Attribute Interfaces
- The FileSystem Class
- The FileSystems Class
- The FileStore Class
- Prospects of NIO
- Working with Streams
- The InputStream Class
- The OutputStream Class
- The ByteArrayInputStream Class
- The ByteArrayOutputStream Class
- The BufferedInputStream Class
- The BufferedOutputStream Class
- The FileInputStream Class
- The FileOutputStream Class
- Working with the Reader Class
- Working with the Writer Class
- Accepting Input from the Keyboard with the InputStreamReader Class
- Working with the OutputStreamWriter Class
- Working with Files
- Using the File Class
- Using the FileReader Class
- Using the FileWriter Class
- Working with the RandomAccessFile Class
- Working with Character Arrays
- Using the CharArrayReader Class
- Using the CharArrayWriter Class
- Working with Buffers
- Using the BufferedReader Class
- Using the BufferedWriter Class
- Working with the PushbackReader Class
- Working with the PrintWriter Class
- Working with the StreamTokenizer Class
- Implementing the Serializable Interface

- Working with the Console Class
- Working with the Clipboard
- Working with the Printer
- Printing with the Formatter Class
- Using the System.out.printf() Method
- Using the String.format() Method
- Formatting Dates Using the String.format() Method
- Using the java.util.Formatter Class
- Scanning Input with the Scanner class
- Summary

Module 7: Implementing Exception Handling

- Overview of Exceptions
- Exception Handling Techniques
- Rethrowing Catched Exception with Improved Type Checking
- Built-in Exceptions
- User-Defined Exceptions
- Summary

Module 8: Working with Multiple Threads

- Using Threads in Java
- Life Cycle of a Thread
- Synchronization of Threads
- Multithreaded Custom Class Loader
- Getting the Main Thread
- Naming a Thread
- Pausing a Thread
- Creating a Thread with the Runnable Interface
- Creating a Thread with the Thread Class
- Creating Multiple Threads
- Joining Threads
- Checking if a Thread Is Alive
- Setting Thread Priority and Stopping Threads
- Synchronizing
- Communicating between Threads
- Suspending and Resuming Threads

- Creating Graphics Animation with Threads
- Eliminating Flicker in Graphics Animation Created Using Threads
- Suspending and Resuming Graphics Animation
- Using Double Buffering
- Simplifying Producer-Consumer with the Queue Interface
- Implementing Concurrent Programming
- Simplifying Servers Using the Concurrency Utilities
- Knowing Various Concurrency Utilities
- Learning about the java.util.concurrent Package
- Learning about the java.util.concurrent.locks Package
- Learning about the java.util.concurrent.atomic Package
- Summary

Module 9: Working with Collections Framework

- The Collection Interfaces
- The Collection Classes
- The Map Interfaces
- The Map Classes
- Collections Framework Enhancements in Java SE 8
- Using the Collection Interface
- The Queue Interface
- The List Interface
- The Set Interface
- The SortedSet Interface
- Using the Collection Classes
- Using the Comparator Interface
- Using the Iterator Interface
- Using the ListIterator Interface
- Using the AbstractMap Class
- Using the HashMap Class
- Using the TreeMap Class
- Using the Arrays Class
- Learning the Fundamentals of Enumerations
- The Legacy Classes and Interfaces
- Using the Aggregate Operations
- Using the java.util.function Package

- Summary

Module 10: Creating Packages, Interfaces, JAR Files and Annotations

- Packages and Interfaces
- JAR Files
- The Core Java API Package
- The java.lang Package
- Basics of Annotation
- Other Built-In Annotations
- Creating a Package
- Creating Packages that have Subpackages
- Creating an Interface
- Implementing an Interface
- Extending an Interface
- Using Interfaces for Callbacks
- Performing Operations on a JAR File
- Marker Annotations
- Single Member Annotations
- Summary

Module 11: Working with Java Beans

- What is Java Bean?
- Advantages of Java Bean
- Introspection
- Persistence
- Customizers
- Understanding Java Beans
- Designing Programs Using Java Beans
- Creating Applets that Use Java Beans
- Creating a Java Bean
- Creating a Bean Manifest File
- Creating a Bean JAR File
- Creating a New Bean
- Adding Controls to Beans
- Giving a Bean Properties
- Design Patterns for Properties

- Using Simple Properties
- Designing Patterns for Events
- Learning Methods and Design Patterns
- Creating Bound Properties
- Giving a Bean Methods
- Giving a Bean an Icon
- Creating a BeanInfo Class
- Setting Bound and Constrained Properties
- Implementing Persistence
- Using the Java Beans API
- Learning the Basics of an Event
- Using the Java Beans Conventions
- Using the Remote Notification and Distributed Notification
- Using Beans with JSP
- Summary

Module 12: Networking and Security with Java

- Basics of Networking
- Sockets in Java
- Client-Server Networking
- Proxy Servers
- Internet Addressing
- Domain Name Service
- Inet4Addresses and Inet6Addresses
- The URL Class
- The URI Class
- URI Syntax and Components
- TCP/IP and Datagram
- Blackboard Assignment Retrieval Transaction
- Understanding Networking Interfaces and Classes in the java.net Package
- Understanding the InetAddresses
- Caching InetAddress
- Creating and Using Sockets
- Creating TCP Clients and Servers
- Understanding the Whois Example
- Submitting an HTML Form from a Java Program

- Handling URL
- Using the URLConnection Objects
- Working with Datagrams
- Datagrams Server and Client
- Working with BART
- Learning about the java.security Package
- Summary

Module 13: Implementing Event Handling and Wrappers in Servlets 3.1

- Introducing Events
- Introducing Event Handling
- Working with the Types of Servlet Events
- Developing the onlineshop Web Application
- Introducing Wrappers
- Working with Wrappers
- Summary

Module 14 : Java Server Pages 2.3 and Expression Language 3.0

- Introducing JSP Technology
- Listing Advantages of JSP over Java Servlet
- Exploring the Architecture of a JSP Page
- Describing the Life Cycle of a JSP Page
- Working with JSP Basic Tags and Implicit Objects
- Working with Action Tags in JSP
- Exploring EL
- Using Custom Tag Library with EL Functions

Module 15: Implementing Filters

- Exploring the Need of Filters
- Exploring the Working of Filters
- Exploring Filter API
- Configuring a Filter
- Creating a Web Application Using Filters
- Using Initializing Parameter in Filters
- Manipulating Responses
- Discussing Issues in Using Threads with Filters
- Summary

Module 16: Java EE Design Patterns

- Describing the Java EE Application Architecture
- Introducing a Design Pattern
- Discussing the Role of Design Patterns
- Exploring Types of Patterns
- Summary

Module 17: Implementing SOA using Java Web Services

- Section A: Exploring SOA and Java Web Services
- Overview of SOA
- Describing the SOA Environment
- Overview of JWS
- Role of WSDL, SOAP and Java/XML Mapping in SOA
- Section B: Understanding Web Service Specifications to Implement SOA
- Exploring the JAX-WS 2.2 Specification
- Exploring the JAXB 2.2 Specification
- Exploring the WSEE 1.3 Specification
- Exploring the WS-Metadata 2.2 Specification
- Describing the SAAJ 1.3 Specification
- Working with SAAJ and DOM APIs
- Describing the JAXR Specification
- JAXR Architecture
- Exploring the StAX 1.0 Specification
- Exploring the WebSocket 1.0 Specification
- Describing the JAX-RS 2.0 Specification
- Exploring the JASON-P 1.0 Specification
- Section C: Using the Web Service Specifications
- Using the JAX-WS 2.2 Specification
- Using the JAXB 2.2 Specification
- Using the WSEE and WS-Metadata Specifications
- Implementing the SAAJ Specification
- Implementing the JAXR Specification
- Implementing the StAX Specification

Syllabus of Spring Course

Module 1: Course Introduction

- Introduction
- Content still to come

Module 2: Install and Setup

- Which Version Of Java?
- Installing JDK 11 on Windows
- Installing and Configuring IntelliJ IDEA on Windows
- Installing JDK 11 on a Mac
- Installing and Configuring IntelliJ IDEA on a Mac
- Installing JDK 11 on a Linux Machine
- Installing and Configuring IntelliJ IDEA on a Linux Machine

New Spring 5:

Module 1: Maven and your first project

- Your Programming Careers Questions Answered
- Access to Programming Career Q&A
- Spring 5 Feature Overview
- Maven and Other Tools
- Creating a Maven Project
- Importing Maven Projects
- Maven Lifecycle Plugin and Goals
- Fix our Maven Project

Module 2: Logging with SLF4J & Logback

- What is Logging?
- Using Logging with Logback
- Logback Configuration

Module 3: Multi module Spring Project

- Project Overview
- Create Multi Module Project
- Project Setup
- Using a Spring Container

- Implementing the Game
- Constructor Based Dependency Injection
- Setter Based Dependency Injection
- Setter or Constructor
- Using Bean Lifecycle Callbacks
- XML or Annotation Configuration
- Autowiring Beans
- Beans as Components
- Using Java Annotation Configuration
- Message Generator Challenge
- Implementing MessageGenerator Logic
- Console Module Setup Challenge
- Application Events
- Finish Game Logic
- Using Qualifiers
- Creating Custom Annotations
- Using Properties
- minNumber Challenge
- Code Cleanup and Constructor Injection
- Code Cleanup Challenge

Module 4: Lombok Introduction

- Lombok Introduction
- Setup Lombok
- Using Lombok
- Lombok Challenge

Module 5: Spring MVC

- Spring MVC Introduction
- Project Setup Challenge
- Setup Maven War Plugin
- Setup Maven Cargo Plugin and Tomcat
- Setup Dispatcher Servlet
- Simple Controller
- View Resolver and View
- Spring MVC Request Processing

- Model and Model Attributes
- Simple Service Challenge
- Request Parameters
- Project Requirements
- To Do Item Class
- To Do Data Class
- Creating the Todo Item Controller
- Creating View and Basic JSTL tags
- Todo Item Service Challenge
- Implementing the Add Item Feature
- Implementing Post Redirect Get Pattern
- Home View and Navigation
- Delete Item Challenge
- Implement Edit Item Feature
- View Item Challenge

Module 6: Spring Boot 2 Introduction

- Introduction to Spring Boot 2
- Using Spring Initializr
- Understanding Spring Boot Project Structure
- Simple Spring Boot Application

Module 7: Spring Boot 2 And Thymeleaf 3

- Add Spring Boot to an Existing Project
- Spring Boot Web Module
- Thymeleaf Introduction
- Setup Thymeleaf and Initial Template
- Spring Boot Developer Tools
- Game Service Challenge
- Play Thymeleaf Template
- Thymeleaf Preprocessing
- Thymeleaf Template Challenge
- Thymeleaf Fragments
- Thymeleaf Fragment Challenge
- Thymeleaf Decoupled Template Logic
- Thymeleaf Decoupled Template Logic Challenge

- Bug Fixes
- Spring Internationalization
- Thmyeleaf Internationalization Challenge
- Message Generator Internationalization Main Message
- Message Generator Internationalization Result
- Request Interception
- Locale Change Interceptor
- Thymeleaf Recap

Module 8: Gradle Introduction

- What is Gradle?
- Creating a Gradle Spring Boot Project
- Gradle Build Lifecycle Projects and Tasks
- Understanding the Gradle Scripts
- Running Gradle Tasks
- Simple Spring Gradle Application

Module 9: Gradle Multi Module Project Setup

- Creating a Spring Boot Project Challenge
- Configure Gradle Multi-module Project
- Configure Sub Modules
- Testing our Project
- More Content

OLD Spring

Module 1: Installation & Setup - Part 1

- Install JDK for Windows
- Install Eclipse for Windows
- Install Tomcat for Windows
- Configure Tomcat Within Eclipse for Windows
- Install JDK for Mac
- Install Eclipse for Mac
- Install Tomcat for Mac
- Configure Tomcat Within Eclipse for Mac
- Install JDK for Linux
- Install Eclipse for Linux
- Install and Configure Tomcat for Linux

Module 2: Introduction to Java EE

- Introduction to JEE
- Exploring Java EE basics: Drilling into the Concepts
- Exploring Java EE basics: Building the Servlet
- Exploring Java EE basics: Creating a JSP and running the App
- Exploring Java EE basics: Adding a Service layer to the mix
- Exploring Java EE basics: Adding JSTL Library Support
- Exploring Java EE basics: JSTL in JSP
- Exploring Java EE basics: Drilling into the Concepts

Module 3: Installation and Setup - Part 2

- Overview of Spring Tooling
- Install Spring IDE
- Installing Maven
- Creating a Maven Project
- Understanding the Project Object Model

Module 4: Introducing Spring Framework

- Why Spring
- Overview of the Spring Framework
- Spring Framework vs JEE
- Introducing the Spring IoC containSpring IoC container: Accessing metadata from the file Spring IoC container: Accessing metadata from theSpring IoC Container Challenge .
- Setting Constructor Arguments using the Spring bean config file
- Setting Properties using the Spring bean config file
- Setter based Dependency Injection in Spring
- Setter based Dependency Injection in Spring continued
- Spring Setter based Dependency Injection in Spring : Running the App
- Constructor based Dependency Injection in Spring
- Spring Constructor based Dependency Injection : Running the App

Module 5: Introducing Spring MVC

- Creating Spring MVC Project Infrastructure
- Adding dependencies in an alternative way in a Spring MVC Project
- Adding Spring MVC Dispatcher Servlet
- Building the Spring Controller - "C" part of MVC

- Context and MVC Configuration
- Autowiring collaborating objects in Spring
- Bringing View Resolver in the mix and running the App

Module 6: Bean Configuration

- Bean scopes for a Spring Bean
- Spring bean scopes: Singleton and Prototype
- Spring bean scopes, Request and Session: creating the project and Java classes
- Spring bean scopes, Request and Session: creating the config file and Controller
- Spring bean scopes, Request and Session: Running the App
- Customize Spring Bean with Callbacks
- Standard Naming for Spring Bean Callbacks
- Spring Bean instantiation using Static Factory and Instance Factory methods
- Spring Bean Definition with C Namespace
- Spring Bean Definition with P Namespace

Module 7: Configuring Spring with Annotations

- Pros and Cons of Spring Configuration : Annotations vs XML
- Autowiring Spring Beans: creating project infrastructure
- Autowiring Spring Beans: using setters, constructors and fields
- Spring Beans as Components: Cutting down on xml config
- Handling Properties in Spring: using XML config
- Handling Properties in Spring: mixing XML with Annotations
- Handling Properties in Spring using pure Annotations: creating infrastructure
- Handling Properties in Spring using pure Annotations: building and running

Module 8: Introducing MySQL

- Install MySQL for Windows
- Install MySQL on Mac
- Install MySQL on Linux
- MySQL Workbench Basics : Environment and table creation
- MySQL Workbench Basics : Insert and Export

Module 9: Working with Spring JDBC

- Introduction to Spring JDBC
- CRUD with Spring JDBC Template: Project Setup
- CRUD with Spring JDBC Template: Creating DAO Layer

- CRUD with Spring JDBC Template: XML config & Property file
- CRUD with Spring JDBC Template: More with DAOs
- CRUD with Spring JDBC Template: DAO and Test Harness
- CRUD with Spring JDBC Template: Building App and executing Test Harness
- CRUD with Spring JDBC Template: Expanding DAO Layer further
- Spring JDBC Exceptions : an Overview
- Spring Named Parameters Template: Understanding the API
- Spring Named Parameters Template: Building the DAO
- Spring Named Parameters Template : Components and Testing

Module 10: Spring MVC in depth - Part 1

- Spring Java Config: Creating the project infrastructure
- Spring Java Config: Creating the Java and Web Configuration
- Spring Java Config: Running the App
- Fix minor error in WebMvcConfig
- JNDI Datasource: Building the Spring Service and Controller
- JNDI Datasource: Conceptual Overview and Configuration
- JNDI Datasource: Building views and Running the App
- Spring MVC Architecture Going Deeper
- Spring Handler Mapping Customization: Creating the infrastructure
- Spring Handler Mapping Customization: Seeing the effect of customization
- Spring Interceptors: an Introduction
- Spring Interceptors: Integrating and building views with CSS inline styling
- Spring Interceptors: Continuing building the views with CSS Internal styling
- Spring Interceptors: Continue building views with mixed styling
- Spring Interceptors: Running the app with a closing summary
- Introducing Logging
- Logging: Working with SLF4J, a powerful logging API
- Logging: LOG4J implementation and introducing Logger, Appender and Layout
- Logging: Continuing with log4j and getting to know Conversion Patterns
- Logging: Configuring log4j.properties and wrapping up
- Spring Interceptors: Invoking all the Interceptor methods
- Spring Interceptors: Running the app and wrapping up!

Module 11: Spring MVC in depth - Part 2

- Section Overview

- RequestMapping and RequestParams In-depth : Introduction & test harness
- RequestMapping and RequestParams In-depth : Class level mapping
- RequestMapping and RequestParams In-depth : Wrapping up first test
- RequestMapping and RequestParams In-depth : method attribute & fallback
- RequestMapping and RequestParams In-depth : defaultAttribute and default naming
- RequestMapping and RequestParams In-depth : resolving ambiguous request mapping
- RequestMapping and RequestParams In-depth : multiple request mapping
- Model Attributes on Methods in-depth: Adding multiple attributes
- Model Attributes on Methods in-depth: working with 'name' attribute
- Model Attributes on Methods in-depth: working with 'value' attribute
- Model Attributes on Methods in-depth: working with ModelAndView API
- Model Attributes on Methods in-depth: Introducing Spring Form Tag and Elements
- Model Attributes on Methods in-depth: Exploring ModelAndView API
- Model Attributes on Methods in-depth: Default data binding
- Model Attributes on Methods in-depth: Implicit resolution of logical view name
- Session Attributes: Creating the project infrastructure
- Session Attributes: Bringing in the Controller
- Session Attributes: Adding the JSPs
- Session Attributes: Cranking the Java configuration
- Session Attributes: Testing our application
- Session Attribute & Request Attribute: an Introduction
- Session Attribute & Request Attribute: Applying to Interceptor and Controller
- Session Attributes, Session Attribute & Request Attribute: Testing all

Module 12:Spring MVC in depth - Part 3 (Forms and Validation)

- Spring Form Tags: An Overview
- Spring Form Tags: Creating project infrastructure - part 1
- Spring Form Tags: Creating project infrastructure - part 2
- Spring Form Tags: SELECT tags - part 1
- Spring Form Tags: SELECT tags - part 2
- Spring Form Tags: SELECT tags - part 3
- Spring Form Tags: CHECKBOX tags - part 1
- Spring Form Tags: CHECKBOX tags - part 2
- Spring Form Tags: CHECKBOX tags - part 3
- Spring Form Tags: CHECKBOXES tags
- Spring Form Tags: RADIOBUTTON & RADIOBUTTONS tags

- Spring Form Validation: an Introduction
- Spring Form Validation: Creating the project Infrastructure
- Spring Form Validation: Creating the JSPs
- Spring Form Validation: Running the first validation test
- Spring Form Validation: Working with @Size and @Notblank validations
- Spring Form Validation: Introducing Custom Validation Constraints
- Spring Form Validation: Creating our first Custom Validation Constraint
- Spring Form Validation: Regular Expression Validation
- Spring Form Validation: Using Apache Commons API Validator
- Spring Form Validation: Class Level Validation
- Spring Form Validation: Final Changes and Test

Module 13: IntelliJ Spring Section

- IntelliJ Spring Config and Test Import
- Import Non Web Based Project
- Build Spring MVC Demo
- Controller Service and JSP
- Welcome Service and Add Beans

OLD – NEW Spring 5

- Install and setup
- Which version of Java should you use?
- How To Install The Java Development Kit (JDK) For Windows
- Download and Install IntelliJ (FREE and PAID version) for Windows
- How To Install The Java Development Kit (JDK) For Mac OS X
- Download and Install IntelliJ (FREE and PAID version) on a Mac
- How To Install The Java Development Kit (JDK) For Ubuntu Linux
- Install and Configure IntelliJ (FREE and PAID version) for Ubuntu Linux

Syllabus of J2EE Course

Module 1: J2EE

- Introduction to JDBC
- Databases and Drivers
- Types of Driver
- Loading a driver class file
- Establishing the Connection to different
- Database with different Driver
- Executing SQL queries by ResultSet,Statements , PreparedStatment interface.
- Using CallableStatement
- Transaction Management & BatchUpdate
- Programs/Interview related Question and Answer

Module 2: JSP

- Basics Of Jsp
- Life cycle of JSP
- JSP API
- JSP in Eclipse and other IDE's
- Programs/Interview related Question and Answer.
- Scripting Elements
- scriptlet tag
- expression tag
- declaration tag
- Implicit Objects
- out
- request
- response
- config
- application
- session
- pageContext
- page
- exception
- Directive Elements

- page directive
- include directive
- taglib directive
- Exception Handling
- Action Elements
- jsp:forward
- jsp:include
- Bean class
- jsp:useBean
- jsp:setProperty & jsp:getProperty
- Displaying applet in JSP
- Expression Language
- What is expression and how to use it
- Define expression and use over the service flow
- The way to be achieve same in JSP
- Mvc In Jsp
- MVC pattern
- Working flow implementation of MVC
- CRUD operation using MVC
- Design a real time web application using MVC
- JSTL
- Discussion on the tag library
- How to implement and use
- Custom Tags
- Custom Tag : What and Why?
- Custom Tag API?
- Custom Tag Example
- Attributes
- Iteration
- Custom URI

Module 3: Servlet

- Basics of Servlet
- Servlet: What and Why?
- Basics of Web
- Servlet API

- Servlet Interface
- GenericServlet
- HttpServlet
- Servlet Life Cycle
- Working with Apache Tomcat Server
- Steps to create a servlet in Tomcat
- How servlet works?
- servlet in Myeclipse
- servlet in Eclipse
- servlet in Netbeans
- Servlet request
- Servlet Request methods
- Registration example with DB
- Servlet Collaboration
- Request Dispatcher
- send Redirect
- Servlet Configure
- Servlet Configure methods
- Servlet Configure example
- Servlet Context
- Servlet Context methods
- Servlet Context example
- Session Tracking
- Cookies
- Hidden Form Field
- URL Rewriting
- HttpSession

Module 4: Concurrent and implementation of collection

- Implemenation of ArrayList
- Implemenation of LinkedList
- Implemenation of HashMap
- Implementation of Queue/PriorityQueue/Deque

Module 5: Advanced Multi-Threading

- Implemenation of Executor pool service and working mechanism with real time

- Big file(Single , multiple) processing using multiple thread
- Implemenation to achieve thread class and runnable interface

Module 6: Javamail Api

- Sending Email
- Sending email through Gmail server
- Receiving Email
- Sending HTML content

Module 7: Design Pattern

- Singleton
- DAO
- DTO
- MVC
- Front Controller
- Factory Method
- Abstract

Module 8: Junit

- JUnit: What and Why?
- Annotations used in JUnit
- Assert class
- Test Cases

Module 9: Maven

- Maven: What and Why?
- Ant Vs Maven
- How to install Maven?
- Maven Repository
- Understanding pom.xml
- Maven Example
- Maven Web App Example
- Maven using Eclipse

Module 10: Project Guidance

Syllabus of Struts Training Course

Module 1: Introduction

- Enterprise
- Enterprise Application
- System logical layers
- Presentation layer
- Business processing layer
- Data Storage and access layer
- System Architecture
- 1-tier Architecture
- 2-tier Architecture
- n-tier Architecture
- Types of EnterpriseApplications
- Web Applications
- Distribute Applications
- WebApplication Models
- Model1-Architecture
- Model2-Architecture
- MVC Architecture& its Rules & Regulations
- FrameWork
- Web Framework
- Application Framework
- Struts Framework History

Module 2: Struts Flow of Execution

Module 3: Struts Elements

- View
- ActionServlet
- RequestProcessor
- FormBean(ActionForm)
- Action class
- web.xml
- Struts Configuration File

Module 4: Struts Tag Library

- Html Tag library
- Bean Tag library
- Logic Tag library
- Nested Tag library
- Tiles Tag library

Module 5: DynaActionForm & LazyDynaBean

Module 7: Validations

- Client Side Validations
- Programmatic Approach
- Declarative Approach
- (Validator Framework)
- Server Side Validations
- Programmatic Approach
- Declarative Approach
- (Validator Framework)

Module 8: Internationalization(I18N)

- I18N at Core level
- NumberFormat
- DateFormat
- ResourceBundle
- I18N at Weblevel(Server & Jsp)
- JSTL format tags
- I18N in Struts

Module 9: Exception Handling in Struts

- Programmatic Approach
- Declarative Approach
- Custom Exceptions in Struts
- Customization on ExceptionHandler

Module 10: Tiles Frame work

Module 11: Built-in Actions in Struts

- IncludeAction
- ForwardAction
- LocaleAction

- DispatchAction
- LookupDispatchAction
- MappingDispatchAction
- EventDispatchAction
- SwitchAction

Module 12: Struts 2.x

- Diff b/w Struts 1.x and Struts 2.x
- Struts 2.x Flow of Execution
- Struts 2.X Elements
- Steps to design Struts Appl. In 2.x version
- Struts 2.x Tag library
- Struts 2.x Application with Annotation
- Struts 2.x Validations

Module 13: Database: Oracle

- Servers:Tomcat & Weblogic
- IDE's: MyEclipse, NetBeans

Syllabus of Hibernate Course

Module 1: Introduction To Hibernate And Concepts

- Need for Hibernate
- Hibernate and ORM (Object-Relation Mapping)

Module 2: Hibernate Configuration

- Required JAR Files
- Hibernate configuration File
- Hibernate properties File
- Hibernate XML File
- SQL Dialects

Module 3: Hibernate Concepts

- Id and Primary Key
- Id Generation Methods
- Transaction

- POJOs (Plain Old Java Objects) and the Data Layer
- Hibernate Over Entity Beans
- Understanding Hibernate Architecture
- Configuration
- SessionFactory
- Session
- Query
- Criteria
- Hibernate Configuration
- Hibernate Mappings
- Persistent Classes
- Working with Hibernate to perform
- CRUD Operations
- Configuring Mappings Using Annotations.

Module 4: Hibernate Mappings

- Component Mapping
- Inheritance Mappings
- Table Per Class Hierarchy
- Table Per Sub Class
- Table Per Concrete Class
- Association Mappings
- One -to-One
- One -to-Many
- Many -to-One
- Many -to-Many

Module 5: Hibernate Query Language

- Select clause
- From clause
- Where clause
- Aggregate functions
- Expressions
- Sorting
- Grouping
- Sub queries

Module 6: Criteria Queries

- Creating Criteria
- Narrowing the Result
- Ordering the Result

Module 7: Native Sql

- Using SQL Query
- Named SQL Query
- Using Stored Procedure for Querying
- Creating Custom SQL for CRUD

Module 8: Hibernate Query Languages And Transactions And Caching

- Using HQL, criteria API, native sql
- Hibernate Transaction
- Hibernate and JDBC
- Hibernate second level caching

Module 9: Spring Hibernate Integrations

- Spring Hibernate Integration
- Data source creation
- Hibernate DAO implementation using

Module 10: Hibernate Uses In Project

- Design Web Application using hibernate
- Hibernate in web application(case studies)
- Project Guidance