Detailed Program of Java Full Stack Developer

Java Full Stack Developer Course Syllabus

Core Java

Introduction To Java

History of Java

What is Java, Java Flavors, characteristics

JVM Architecture

Bytecode

Class Loader

Unicode

Class path

Path

Fundamentals of Java Programming

Object oriented concepts (OOP)

Keywords, Datatypes, Variables, Operators, Casting

Selection statement (if, switch)

Control statements (while, do while , for)

Conditional statements (if, else, elseif, ?

static

Arrays

Object Oriented Programming with Java

Classes and Objects

Structure of a class – its internals (Data Members, methods)

Using static

Constructor

this keyword

modifiers

playing with the object (copying, casting)

Garbage collection

Abstract class

Inheritance

Basics (extends keyword)

Modifiers and their scope

Deriving a class

super, final keyword

why java does not support multiple inheritance?

Polymorphism

overloading a method

overloading a constructor

method overriding

accessing base class method

Packages and Interfaces

basics

modifiers and their scope chart

setting classpath

compiling and accessing a packaged class

types of packages

user defined package

Exploring java.lang package

String, StringBuffer, Arrays,

Wrapper classes

Exception Handling

Basics

Hierarchy of exceptions

Handling exception – Try, catch, finally, throw, throws

User defined exceptions

Threads

Basics, Thread class, Runnable Interface

Thread model

Life cycle – start(), run()

Scheduling

Deadlocks / concurrency issues

Synchronization – as a block, as a modifier

Daemon thread

I/O Streams

Introduction

Hierarchy of streams

IO Stream, Byte Stream, Character Streams

BufferedInputStream, BufferedOutputStream

Reader and Writer class

BufferedReader, PrintWriter

Serialization

Collection Framework

basics, hierarchy

legacy classes – Vector, Queue, Stack, Enumeration, Dictionary, Properties

List, ArrayList, LinkedList

Set, HashSet, TreeSet,

Map, HashMap, TreeMap,

Generics

Annotations

Boxing / Unboxing

Enums

Introduction to functional style of programming

Mini application – discussion / implementation

Tools

Javadoc

javap

jar

IDE Tools

eclipse

myeclipse

****

Spring

Spring core

Types of containers

Xml configuration

Core annotations

Component – scan

What can be injected into a bean

Understanding core annotations

Spring MVC

Handler Mapping

Controllers

View resolvers

Validators

interceptors

Spring DAO

Jdbc templates

Exception hierarchy

Spring ORM

Hibernate template

Integration with hibernate

Spring AOP

Point cut

Advisors

Types of advices

Spring – webservices (Restful API)

Spring security

Spring JPA

****

Hibernate

Introduction

Introduction to ORM (Object Relational Management)

Benefits of ORM

Contrast JDBC with ORM

Hibernate Jar files

Hibernate Architecture

SessionFactory , Session, Transaction, Query

Understanding config files .hibernate.xml, .hbm.xml

Performing CRUD operations

load, get, save, saveorupdate, delete

HQL

working with various SQL commands

accessing more than one object from the table

Mapping

Inheritance

Collection

Composition

Association

Cache

Level – I and II

Using third party to configure L2 cache

Criteria

Interfaces / classes

implementation

Note : an application using the above technologies –

Domains : Telecom , LMS, etc…

****

Spring Boot

Introduction / benefits

Spring Boot overview

Bootstrapping an Boot application –

Initializer

Auto Configuration

Spring Boot annotations

Spring Boot properties

Spring Boot profiles

Accessing data with Spring Boot and h2 database

Configuring Spring Boot with spring MVC application

Building a Restful web application with Spring Boot

Highlights of Training

An application based on Spring boot – Either migration / new one

Industry experienced Professional

Hands-on experience with Project orientation

Interview based Questions

Advanced Java Training Syllabus

Fundamentals

Client, Server, Port, Application, Web Server, Application Server, Web Client and Web Application

Installation of Java and Understanding PATH and CLASSPATH

Installation of Eclipse

Installation of Tomcat and structure of Tomcat

Linking any Web Server with Eclipse

Static Resource Access from Tomcat

Servlets

Servlet Internals and Servlet Programming

What are Servlets and why do we need Servlets?

Servlet Development with Eclipse

– Developing Servlets

– Compiling Servlets

– Deployment and Invoking of Servlets

Purpose of Servlet Mapping and web.xml

Servlet Architecture

– Detailed Architecture of Servlets

– Discussion of GenericServlet, HttpServlet Interfaces

– Programming examples

Servlet Life Cycle

HTTP Internals

Detailed discussion of all parameters of request and response packets

Discussion of HttpServletRequest and HttpServletResponse interfaces

Session Tracking API

– Practical visualization of Http Session objects and Session table

– Session Tracking API

– Session Tracking Programming

Cookies, URL Rewriting and Hidden Form Fields

ServletConfig Object

ServletContext Object

RequestDispatcher Object

Filters

– Why do we need filters?

– Filter, FilterChain, FilterConfig interfaces

– Programming examples

Application / Attribute Lifecycle Listeners

Why do we need Lifecycle Listeners?

ServletContext and ServletContext Attribute Listeners

Session and Session Attribute Listeners

Programming examples

JSP

Java Server Pages(JSP) Internals and Programming

What is JSP? Why do we need JSP?

JSP development with Eclipse

JSP Life Cycle

Predefined Objects of JSP

request, response, out and session objects

config, application, pagecontext objects

Scripting Tags (Writing Java code inside JSP)

Directives

@page, @include directives

Action tags of JSP

Difference between jsp:forward, jsp:include and @include tags

Programming examples to illustrate each tag

Expression Language

Why do we need EL?

Reading scoped attributes using EL

Reading bean properties using EL

Reading collections using EL

Operators of EL

Programming examples to illustrate each usage

JSTL (JSP standard tag library)

Why do we need JSTL?

Looping tags

Control tags

Database tags

118N tags

Programming examples to illustrate each category of tags

Custom tags

Custom tag creation based on Servlets

Custom tag creation based on JSP

concept of tld file

programming examples to illustrate each concept

Java Beans

What are Java beans and why do we need Java beans?

Action tags for java beans

Application of Java beans

MVC Architecture

Why do we need MVC?

Discussion of pros and cons of Servlets, JSP and JavaBeans

Programming example to illustrate Business logic, Controller logic, Presentation logic and Bean creation

JDBC

Java Data Base Connectivity (JDBC) internals and programming

What is JDBC ? Why do we need JDBC ?

Discussion of JDBC driver types and Driver Manager

Connection object

Statement types and Discussion of each type of Statement Objects

ResultSet Types and Discussion of each type

Stored Procedure Access from java applications

Transaction Programming with JDBC

Savepoint and RowSet

DataSource and Connection Pooling

Batch Updates

JNDI

Explanation on Usage of important Practical Tools

Eclipse

Jboss, Weblogic, Tomcat

****

Database

Introduction to MySQL

MySQL Installation

DDL Commands

DML Commands

Operators

Functions

Constraints

Joins

Views

Indexes

DCL Commands

TCL Commands

MySQL Cursors

Exception Handling

Stored Procedure

Stored Functions

****

MicroServices

MicroServices Introduction

Principle and Characteristics

Use cases and Benefits

Challenges

Design standards

Micro Services Communication

Pitfalls

****

UI Development

HTML 5

CSS 3

JavaScript

Advanced JavaScript

BootStrap

Angular

jQuery

Ajax

****

DevSecOps Tools

Docker

Agile with Jira

Chef

LOG4J / SLF4J

GitHub

Gradle

Maven

ELK

Heroku

SonarQube

Amazon Cloud Watch

Datadog

****

Cloud – Application Deployment on Cloud

AWS Cloud

What is AWS

Introduction to AWS Services

Amazon Cloud Services

Create EC2 Instance in AWS

Deploy Spring Boot Application on AWS | Elastic Beanstalk

AWS Advanced Research and Development- Live Deployment