



IT - FINISHING SCHOOL
ISO 9001 : 2008 CERTIFIED

COURSE SYLLABUS

J2EE Technologies

Industrial Training

(4 MONTHS)



PH: 0481 2411122, 09495112288
E-Mail: info@faithinfosys.com

Marett Tower
Near No. 1 Pvt. Bus Stand
Vazhoor Road
Changanacherry-01

www.faithinfosys.com

Target Audience

This course is specially designed for the B.Tech/B.E, M.Tech/M.E and all other IT related Graduates and Post Graduate students who are interested in learning hottest programming language JAVA.

Mission

Professionalism has conquered the job scenario and companies seek for well qualified, professional and skilled manpower. Keeping in view this demand of companies we groom students in such a way that they will be second to none. Quality Education and Performance Oriented Training is our motto.

Course Overview

This course covers extensively the programming and object-oriented techniques of Java. It introduces students to fundamentals of the Java language and syntax, major class libraries and prepares them to begin development of business applications in Java. It is interspersed with step-by-step exercises illustrating the concepts as they are explained.

Live Project Work

Live project is the phase when you finally implement most of the things that you have learnt during your software training. Software development is more than just coding. Before you write even a single line of code, it requires careful analysis of the requirements, gathering information, preparing the necessary documentation which requires understanding the live project using Software Development Life Cycle. So you have to learn tricks to produce bulk output on time maintaining the right design quality or coding standard. That is the significance of Live Project Training. We assure that our Live Project Training will impart the confidence in students to work on real time projects.

Add on Programs

- Personality Development
- Listening Skills
- Communication Skills
- Interview Skills
- Group Discussion

- Topics Presentation
- Awareness of IT Trends
- Aptitude Tests
- Technical Tests
- Mock Interview

SOFTWARE ENGINEERING WITH UML

This course covers concepts of software engineering. It intends to lay a foundation for software designing and professional practice by conveying fundamental knowledge about software development process, requirements analysis, design techniques, and testing methods. The course emphasizes on modeling skills with the Unified Modeling Language (UML).

Section 1: Overview of Software Development

- 1.1 Software Engineering Concepts
- 1.2 Software Engineering Development Activities
- 1.3 System Development Models and Approaches
- 1.4 Software Process and Project Management

Section 2: Software Development Life Cycle

- 2.1 Requirement Elicitation
- 2.2 Analysis
- 2.3 System Design
- 2.4 Object Design
- 2.5 Implementation
- 2.6 Testing

Section 3: UML

- 3.1 Introduction
- 3.2 Need of UML
- 3.3 Use Case Driven Object Oriented Analysis
- 3.4 Use Case Model
- 3.5 Use Case Diagram
- 3.6 Activity Diagram
- 3.7 Sequence Diagram
- 3.8 Collaboration Diagram
- 3.9 Class Diagram

Section 4: Project Development models

- 4.1 Waterfall
- 4.2 V model
- 4.3 Prototype model
- 4.4 Spiral model

****Java****

Course Description:

The contents of this course are a comprehensive solution that moulds you to a java specialist by providing a combination of on hand labs and the training provided in the class. It helps the trainee to learn and develop various java technology applications that definitely meets the current industry needs.

CORE JAVA

Section 1: Object Oriented Programming - How to Approach a System?

- 1.1 Describe the concept and features of object-oriented programming
- 1.2 Create classes and objects and add methods to a class
- 1.3 Basic concepts in OOPS
- 1.4 Real World Comparison

Section 2: Evolution of JAVA

- 2.1 Know the history of Java
- 2.2 Requirements and Environment(JDK)
- 2.3 Comparison with other languages
- 2.4 Basic Features

Section 3: Learn the Fundamentals of Java Programming

- 3.1 Understanding Java Architecture
- 3.2 Understanding Java Virtual Machine(JVM)
- 3.3 Understanding and Installing Java Development Kit
- 3.4 Knowing the Program Structure
- 3.5 Data types , Variables and Operators
- 3.6 How to handle arrays in Java
- 3.7 Dealing with various Programming Constraints

Section 4: Leaping into java - The Logics and Techniques

- 4.1 Managing Classes and Objects in Java
- 4.2 Managing Functions
- 4.3 Understanding Constructors
- 4.4 Dealing with Garbage Collection
- 4.5 Working with Inheritance in Java
- 4.6 Understanding Abstract Classes and Interfaces

Section 5: Packages - The Library Management

- 5.1 Introduction to packages
- 5.2 Know How to implement a package
- 5.3 CLASSPATH Setting for Packages.
- 5.4 Making JAR Files for Library Packages
- 5.5 Import and Static Import
- 5.6 Knowing java.lang package
- 5.7 Knowing java.util package
- 5.8 Knowing java.io package (input/output programming)
- 5.9 Streams and the new I/O Capabilities
- 5.10 Understanding Streams
- 5.11 The Classes for Input and Output

Section 6: Multithreaded Programming - The Inside Parallelism

- 6.1 Basic concepts and needs of multi-threading
- 6.2 Life Cycle of a Thread
- 6.3 How to create a thread
- 6.4 Handling Thread Priorities
- 6.5 Enforcing Thread Synchronization
- 6.6 Maintaining Interthread Communication
- 6.7 Other thread functions

Section 7: Exception Handling - The Error Management

- 7.1 The Idea behind Exception
- 7.2 Exceptions & Errors
- 7.3 Types of Exception
- 7.4 Control Flow In Exceptions
- 7.5 JVM reaction to Exceptions
- 7.6 Use of try, catch, finally, throw, throws in Exception Handling

Section 8: Applet Programming - The Web Component Development

- 8.1 Applet Basics
- 8.2 Applet Architecture
- 8.3 Parameters to Applet
- 8.4 Embedding Applets in Web page
- 8.5 Creating Applet applications

Section 9: Event Handling - The Component's Response

- 9.1 The Delegation Event Model
- 9.2 Event Classes
- 9.3 Event Listener Interfaces
- 9.4 Handling Various Events

Section 10: GUI Programming - The Easiness of Interaction

- 10.1 Introduction to AWT
- 10.2 Know the Window Fundamentals
- 10.3 Working with Frame Windows
- 10.4 Working with Graphics
- 10.5 Using AWT Controls and Menus
- 10.6 Understanding Layout Managers

Section 11: JFC and Swing - A Higher Level of User Interaction

- 11.1 Features of the Java Foundation Classes
- 11.2 Overview of Swing
- 11.3 Components and Containers
- 11.4 Swing Packages
- 11.5 Exploring Swing components
- 11.6 Generating Swing Application

Section 12: Database Connectivity using JDBC

- 12.1 Understanding JDBC
- 12.2 Define the layers in JDBC architecture
- 12.3 Various types of JDBC drivers
- 12.4 Manipulating various SQL Queries
- 12.5 Manage transactions and perform batch updates in JDBC
- 12.6 Creating Database Connectivity Applications

Section 1: HTML5- The Static Web Page Creation

- 1.1 HTML5 Introduction
- 1.2 Structure
- 1.3 Elements
- 1.4 Semantics
- 1.5 Audio & Video
- 1.6 Section & Article
- 1.7 Canvas, Aside
- 1.8 Drag & Drop
- 1.9 Forms & Form Elements

Section 2: CSS 3 - The Presentation Semantics

- 2.1 CSS Properties, Selectors, Style Declaration Types
- 2.2 Colors, Backgrounds, Text and Fonts
- 2.3 Images, Links, Tables and List
- 2.4 Borders, Padding, Margin
- 2.5 Cursor, Dimension, Scrollbars, Visibility and Positioning
- 2.6 Pseudo class & Elements, @Rules(import, font-face, charset)
- 2.7 Filters, Media Types, Printing and Layouts

Section 3: JavaScript - The Interpreted Programming Language

- 3.1 Interpreted Programming Languages
- 3.2 Integrating JavaScript with HTML
- 3.3 Variables in JavaScript
- 3.4 Operators in JavaScript
- 3.5 Expressions in JavaScript
- 3.6 Arrays in JavaScript
- 3.7 Handling Loops & Decision structures
- 3.8 Executing Conditional statements
- 3.9 Working with Functions

Section 4: JQuery - Write Less Do More...

- 4.1 Understanding jQuery
- 4.2 jQuery Selectors
- 4.3 Event Manipulation Methods
- 4.4 Sliding, Easing, Fading, Toggling
- 4.5 jQuery and AJAX calls
- 4.6 JSON

Section 5: Bootstrap

- 5.1 Introduction to Bootstrap
- 5.2 Bootstrap Grid System
- 5.3 Creating Layouts with Bootstrap
- 5.4 Bootstrap CSS - Understanding the CSS
- 5.5 CSS Customization / Skins
- 5.6 Responsive Web design with Bootstrap
- 5.7 Single Page Responsive site with Bootstrap
- 5.8 Bootstrap Layout Components
- 5.9 Bootstrap Plug-ins :
 - ✓ Transition
 - ✓ Modal
 - ✓ Dropdown
 - ✓ Scrollspy
 - ✓ Tab
 - ✓ Tooltip
- 5.10 Building Websites with Bootstrap

Section 6: AngularJS

- 6.1 AngularJS Introduction
- 6.2 Single Page Application (SPA)
- 6.3 Directive, Filters and Data Binding
 - ✓ What are Directives?
 - ✓ Using Directives and Data Binding Syntax
 - ✓ Data-Binding Example using AngularJS Directives
 - ✓ Iterating with the ng-repeat Directive
 - ✓ ng-repeat Example
 - ✓ The AngularJS API Reference for Directives
 - ✓ Using Filters
 - ✓ Using Filters Demo
- 6.4 Views, Controllers and Scope
- 6.5 Modules, Routes And Factories
 - ✓ Creating a Module
 - ✓ Creating a Controller in a Module
 - ✓ The Role of Routes
 - ✓ Defining Routes
 - ✓ Defining Routes Demo
 - ✓ Using Factories and Services
 - ✓ The Role of the Factory

*** (All modules of Core Java +...)*

Section 1: Web Application Development (Using Servlets)

- 1.1 Introducing Web Concepts
- 1.2 Knowing Web Servers
- 1.3 Introduction to Servlet
- 1.4 Servlet Life Cycle
- 1.5 The Servlet API
- 1.6 Now the Session Management Techniques
- 1.7 Managing Inter Servlet Communication
- 1.8 Servlet Filters

Section 2: JSP Web Application Development

- 2.1 Introducing JSP
- 2.2 Structure of JSP Programming
- 2.3 Knowing Scripting Elements
- 2.4 Knowing JSP Directives
- 2.5 Developing a JSP Application
- 2.6 Create robust web applications using session management and database integration

Section 3: XML - The Data Carriers

- 3.1 Knowing XML
- 3.2 Understanding XML Structure
- 3.3 XML Document Type Definitions(DTD)
- 3.4 XML Schemas and Parsers
- 3.5 XML Based API's
- 3.6 Application Development using XML and DTD

Section 4: Business Component Development with JavaBeans Technology

- 4.1 Basic Concepts
- 4.2 Manifest file and JAR file
- 4.3 Developing a JavaBean
- 4.4 Implementing JavaBeans

Section 5: Networking - Managing the Distributed System

- 5.1 Networking Basics and Concepts
- 5.2 Various Networking Protocol Awareness(TCP/IP, UDP)
- 5.3 Networking Classes and Interfaces
- 5.4 Knowing InetAddress and URL
- 5.5 Datagram and Socket Programming
- 5.6 An Application Development

Section 6: Remote Method Invocation

- 6.1 Understanding RMI
- 6.2 The RMI system architecture in detail.
- 6.3 What are Stubs and Skeletons
- 6.4 Understanding RMI Packages
- 6.5 Creating an RMI application

*** (All modules of Core Java and Advanced Java + ...)*

Section 1: An Overview of J2EE

- 1.1 Introducing J2EE concepts
- 1.2 Understanding the 3- Tier Scenario
- 1.3 Working with Model-View-Controller
- 1.4 Introducing Application Servers

Section 2: Java Mail API - Building Mail Applications

- 2.1 Understanding the protocols for JavaMail
- 2.2 Knowing the JavaMail Architecture
- 2.3 Working with JavaMail API

Section 3: Java Messaging Service - Building Message Applications

- 3.1 Understanding JMS Architecture
- 3.2 Understanding Messaging Models
- 3.3 Understanding JMS Components
- 3.4 JMS API Programming Model & Examples
- 3.5 Introducing Message Driven Beans

Section 4: Web Services - Sharing the Web

- 4.1 Defining Web Services
- 4.2 Technologies behind Web Servers
 - SOAP
 - WSDL
 - UDDI
- 4.3 J2EE Web Services

Section 5: Enterprise Java Beans - The Backend Business Code

- 5.1 Learn EJB Component Technology
- 5.2 Understand EJB Architecture
- 5.3 Explaining Session Beans
- 5.4 Handling Entity Beans
- 5.5 Understand CMP & BMP
- 5.6 How to use Message Driven Beans

Section 6: More J2EE Techniques

- 6.1 Examining JNDI and Directory Services
- 6.2 Java Transactions

*** (Strong Background in Core Java, Servlets, JSP and JDBC +...)*

Section 1: Struts - An MVC Design Para-diagram

- 1.1 MVC Architecture Model
- 1.2 Type2 MVC
- 1.3 Struts Architecture and Configuration
- 1.4 Form processing
- 1.5 Struts tags
- 1.6 Other Features of Struts

Section 2: Spring - The Super Container of Java Objects

- 2.1 Lightweight and Heavyweight Containers
- 2.2 Inversion of Control
- 2.3 Spring Containers
- 2.4 Simple Examples
- 2.5 Spring DAOs
- 2.6 Spring MVC
- 2.7 Spring Hibernate
- 2.8 Spring Struts
- 2.9 Struts-Spring-Hibernate

Section 3: Hibernate - The Powerful ORM Library

- 3.1 Overview of ORM network
- 3.2 Installing and configuring Hibernate
- 3.3 Overview of Hibernate Framework
- 3.4 Basic CRUD Operations
- 3.5 Hibernate Annotations
- 3.6 Hibernate Query Language

Oracle

Section 1: Relational Database Basics

- 1.1 Brief History of Oracle
- 1.2 Relational Databases and Popular Databases
- 1.3 Oracle Statements

Section 2: Data Manipulation Language (DML)

- 2.1 INSERT
- 2.2 UPDATE
- 2.3 DELETE
- 2.4 SELECT

Section3: Data Definition Language (DDL)

- 3.1 CREATE
- 3.2 ALTER
- 3.3 DROP

Section4: Subqueries, Joins and Unions with Oracle

- 4.1 Order By, Like , And & Or, Where, Between
- 4.2 Joins & Unions
- 4.3 Aggregate Functions and Grouping

Section 5: Constraints and Normalization

- 5.1 Understanding Primary and Foreign Keys
- 5.2 Understanding Database Normalization
- 5.3 Views and Triggers

Live Project (The Mapping of what you learn...)

A software development process provides a basis for the organized production of software, using a collection of predefined techniques and notations. The process starts with the formulation of the problem, and then continues through analysis, design and implementation.

Development Stages:

1. System Conception

Conceive an application and formulate tentative requirements. It deals with genesis of an application.

2. Analysis

Analysis focuses on creation of models. It specifies what must be done, not how it should be done. Developers must fully understand the problem before addressing the additional complexities of design. During analysis developers consider the available sources of information and resolve ambiguities.

3. System Design

During system design, the developer makes strategic decisions with broad consequences. The system designer must understand how a new system interacts with other system, and the system must support future modifications

4. Class Design

Developer expands and optimizes analysis models; there is a shift in emphasis from application concepts toward computer concepts. Developers choose algorithms to implements major system functions.

5. Implementation

Implementation is the stage for writing the actual code. Developers map design elements to programming language and database code.

6. Testing

During testing, Testers once again revisit the original business requirements and verify that the system delivers the proper functionality. It also uncovers the accidental errors that have been introduced.
