



COURSE SYLLABUS

ANDROID

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 development using ANDROID.

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 Android course is designed to introduce and familiarize participants with programming in the Android environment. It starts with an examination of the core java components and concepts that define the Android platform, and then moves on to cover the specific structure that comprises an Android application. The unique characteristic of programming in the mobile environment are introduced and 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

**** 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, especially in the Android Developing Environment.

CORE JAVA

Section 1: Object Oriented Programming

- 1.1 OOPS fundamentals
- 1.2 Classes and Objects
- 1.3 Other concepts in OOPS
- 1.4 How to apply

Section 2: Evolution of JAVA

- 2.1 History of Java
- 2.2 Requirements and Environment(JDK)
- 2.3 Comparison with other languages
- 2.4 Basic Feature

Section 3: Development Fundamentals

- 3.1 Java Architecture
- 3.2 Java Virtual Machine
- 3.3 Program Structure
- 3.4 Data types , Variables and Operators
- 3.5 Array Handling in Java
- 3.6 Programming Constraints

Section 4: Leaping into java

- 4.1 Classes and Objects in Java
- 4.2 Functions
- 4.3 Constructors
- 4.4 Garbage Collection
- 4.5 Inheritance in Java
- 4.6 Abstract Classes and Interfaces

Section 5: Packages

- 5.1 Introduction to packages
- 5.2 Creating and Importing a Package
- 5.3 Knowing java.lang package
- 5.4 Knowing java.io package (input/output programming)
- 5.5 Knowing java.util package

Section 6: Multithreaded Programming

- 6.1 Basic concepts
- 6.2 Life Cycle of a Thread
- 6.3 Creating a thread
- 6.4 Thread Priorities
- 6.5 Synchronization
- 6.6 Interthread Communication
- 6.7 Other thread functions

Section 7: Exception Handling

- 7.1 Understanding Exceptions
- 7.2 Types of Exceptions
- 7.3 Handling Mechanism

Section 1: Introduction to Android

- 1.1 Overview of Android - An Open Platform for Mobile development
- 1.2 Why to use Android for mobile development?
- 1.3 Android Versions and Features
- 1.4 Android API Levels

Section 2: ANDROID Internals

- 2.1 Android Architecture
- 2.2 Dalvik Virtual Machine
- 2.3 JVM Vs DVM Comparison
- 2.4 Basic Building Blocks

Section 3: Android Application Structure

- 3.1 Introduction
- 3.2 Understanding Intent
- 3.3 Activities
- 3.4 Android Activity Lifecycle
- 3.5 Services
- 3.6 Broadcast Receivers
- 3.7 Content Providers
- 3.8 Android Manifest
- 3.9 Creating First Android Application

Section 4: Emulator-Android Virtual Device

- 4.1 Launching emulator
- 4.2 Editing emulator settings
- 4.3 Emulator shortcuts

Section 5: Android UI Design

5.1 Simple UI -Layouts and Layout properties

- 5.1.1 Fundamental Android UI Design
- 5.1.2 Introducing Layouts
- 5.1.3 Creating new Layouts
- 5.1.4 Relative Layouts
- 5.1.5 Table Layouts
- 5.1.6 Frame Layout
- 5.1.7 Linear Layout
- 5.1.8 Nested Layout

5.2 XML Introduction to GUI objects viz

5.3 GUI Elements

5.4 Examples

Section 6: Android user interface widget

- 6.1 Push Button
- 6.2 Basic widgets
- 6.3 Text / Labels
- 6.4 Edit Text
- 6.5 Toggle Button
- 6.6 Weight Sum
- 6.7 Padding
- 6.8 Layout Weight
- 6.9 Android - Event Handling

Section 7: Menu, Dialog, Toast

- 7.1 What is Menu?
- 7.2 Creating and Using Menu
- 7.3 Know Dialogs
 - 7.3.1 Different Dialogs
- 7.4 Toast Notifications
 - 7.4.1 Introducing Toasts
 - 7.4.2 Basic Toast Notification
 - 7.4.3 Positioning Toasts
 - 7.4.4 Custom Toasts
- 7.5 List & Adapters

Section 8: Styles & Themes

- 8.1 Styles.xml
- 8.2 Colors.xml- declaring colors and drawables
- 8.3 Drawable resources for shapes, gradients(selectors)
- 8.4 Style attribute in layout file
- 8.5 Applying themes

Section 9: Knowing Intents

- 9.1 Intents
- 9.2 Intent filter
- 9.3 Creating & Handling Intents
- 9.4 Explicit Intents
- 9.5 Implicit intents
- 9.6 Examples

Section 10: Threads

- 10.1 Threads running on UI thread
- 10.2 Worker thread
- 10.3 Handlers & Runnable
- 10.4 Examples

Section 11: Multimedia Programming using Android

- 11.1 Multimedia audio formats - Creating and Playing
- 11.2 Multimedia audio formats - Kill / Releasing (Memory Management)
- 11.3 How to associate audio in any application
- 11.4 How to associate video playback with an event

Section 12: Persistent storage in Android

- 12.1 Preferences
- 12.2 Introducing SQLite Database
- 12.3 SQLite Open Helper and creating a database
- 12.4 Opening and closing a database
- 12.5 Working with cursors Inserts, updates, and deletes
- 12.6 File I/O

Section 13: Location Based Services and Google Maps

- 13.1 Using Location Based Services
- 13.2 Finding current location and listening for changes in location
- 13.3 Geocoder
- 13.4 Working with Google Maps
- 13.5 Displaying route on map

Section 14: Animation

- 14.1 Android Animation API
- 14.2 Android Drawable class
- 14.3 Android Animation Example
- 14.4 Android Rotate Animation
- 14.5 Android Fade Animation
- 14.6 Android Zoom Animation

Section 15: Device Connectivity

- 15.1 Bluetooth Tutorial
- 15.2 List Paired Devices
- 15.3 Working with WiFi
- 15.4 Working with Camera

Section 16: Android Graphics

- 16.1 Graphics API
- 16.2 2D Graphics
- 16.3 android.graphics.Canvas
- 16.4 android.graphics.Paint class

Section 17: Sensor

- 17.1 Sensor API Motion Sensor
- 17.2 Position Sensor
- 17.3 Environmental Sensor
- 17.4 Sensor Values
- 17.5 SensorManager class
- 17.6 Sensor Class
- 17.7 SensorEvent class
- 17.8 SensorEventListener interface
- 17.9 Compass Accelerometer and Orientation Sensors

Section 18: Testing on Real Devices

- 18.1 Deploying an Android application on physical device
- 18.2 Example illustration
- 18.3 Renew and recap

SQL

Section 1: Relational Database Basics

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

Section 2: Data Manipulation Language (DML)

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

Section 3: Data Definition Language (DDL)

- 3.1 CREATE
- 3.2 ALTER
- 3.3 DROP

Section 4: Sub-Queries, Joins and Unions with MySQL

- 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

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.
