TechyEdz Solutions

A Blended Learning Approach

Android App Developer

☎ 9032803832

9032803832

contact@techyedz.com

www.techyedz.com

Android App Development

Course Overview:

Starting your career in Android App Development Course will help you learn the fundamentals of Kotlin, Google's preferred programming language for Android and build interesting applications. The course will cover the Android components, tools and technologies like multi-screen Navigation, Android Studio, Intents, Fragments, Widgets, Layout and Ionic to build modern applications. The course will enable you to use architectural techniques to build user-friendly and interactive applications.

What you'll learn

- Make pretty much any Android app you like (your only limit is your imagination)
- Submit your apps to Google Play and generate revenue with Google Pay and Google Ads
- Become a professional app developer, take freelance gigs and work from anywhere in the world
- Bored with the same old, same old? Apply for a new job in a software company as an Android developer

Course Outline:

Introduction

- Why Android?
- Key Advantages
- Course Overview
- Interesting Facts about this Course
- Evolution of Android Application Development
- Establishing the Need
- Market Trends and Predictions for Android Apps
- Objectives
- Course Breakdown

Introduction to Android Development

- Installing Android SDK and Android studio
- Android Architecture
- Key Features of Android 6.0 Marshmallow
- Creating Android Project
- Simple App Demo to take input and say "hello + name"

Android Activity and Intents

- Activity Life Cycles
- Intents, Passing data, Launching activities
- Supporting different devices
- Adding Action Bar
- Saving data

Layouts and Controls

- Android Layouts
- Menus
- Input controls
- Settings
- Toasts
- Drag and Drop

List Views and SQLite

- List Views
- Loader
- Methods to manage SQLite database

Services

- Bound services
- Creating a bound service
- Managing a lifecycle of bound service
- Communicating with remote service

Content Providers

- Content provider basics
- Creating content providers
- Calendar provider
- Contact provider

UI and Fragments

- Fragments
- Dynamic UI with Fragments
- Loaders
- Tasks and back stack

Location and Sensors

- Getting a current location
- Using maps with location
- Location strategies
- Sensors overview
- Motion sensors
- Position sensors
- Environment sensors

Multimedia Audio and video & camera

- Media playback
- Media router
- Media route provider
- Camera

Connecting Devices wirelessly

- Android Wireless APIs
- Network Service Discovery
- Registering the Service
- Discovering Services on Network
- Resolving a Service on the Network
- Unregistering a Service
- P2P Connections with Wi-Fi
- Wi-Fi P2P Connection Setup
- Wi-Fi P2P for Service Discovery

Network Operations

- Connecting to the Network
- Sending a Request and Receiving a Response
- URL Connection
- XML Data
- Parsing XML

Network Operations

- Connecting to the Network
- Sending a Request and Receiving a Response
- Media route provider
- Managing Network Usage
- XML Data
- Parsing XML

Battery Optimization

- Optimizing Downloads
- Optimizing Downloads—Prefetching Data
- Minimizing Regular Server Updates

Syncing with cloud/Server

- Syncing with Cloud
- Auto Backup in Android 6.0
- Supporting Older Versions of Android
- Sync Adapters
- Sync Adapters—Components
- Creating a Stub Authenticator
- Creating a Stub Content Provider

Using Volley for Network Data interaction

- Volley—Introduction
- Getting Volley Library
- Using Volley
- Setting up a Network Queue
- Standard Requests
- Implementing Custom Request

Building App for enterprise

- Android for Work
- Managed Profiles
- Compatibility with Managed profiles
- Testing Compatibility with Managed Profiles
- Android for work—Application Restrictions

Google Play to distribute and monetize

- Selling In-App Products
- Preparing In-App Billing
- Creating In-App Products
- Query In-App Products
- Purchasing In-App Product
- Consuming In-App Purchased Product

User Interface

- Adaptive UI Flows
- Handling Screen Configuration
- App Bar
- App Bar—Implementation
- Adding Actions
- Action Bar—Up Navigation

Security & privacy

- Security and Privacy—Overview
- Storing Application Data in Files
- Storing and Sharing Application Data
- Using and Creating Permissions
- Security Tips for Networking
- Using HTTPS and SSL—Overview

Testing

- Test Project in Android Studio
- Android Application Testing—Overview
- Instrumentation—Overview
- Test Case Classes
- Assertion Classes and Mock Objects
- Running Tests and Getting Results
- Activity Testing
- Espresso for UI Testing

User Input

- Detecting Common Gestures
- Detect and Handle Touch Gestures
- Tracking Movement
- Handling Multi-Touch Gestures
- Handling Drag or Scroll gesture
- Scaling

Creating Wearable Apps

- Android Wear Introduction
- Android Wear User Interface principles
- Android Wear Application design principles
- Android Wear Application types

- UI Patterns for Android Wear - Cards
- UI Patterns for Android Wear - Pages

Custom UI for Wearable’s

- Defining Layouts
- Defining Layouts using Watch View Stub element
- Accessing Layout Views in Watch View Stub
- element
- Using Shape-aware layout
- Adding Cards
- Creating Lists
- Creating a 2D Picker

Adding wearable features

- Adding Wearable Features to Notifications
- Creating Notifications for Wearable’s
- Adding Buttons to Notifications
- Adding Big View to Notification
- Adding Wearable Features

Data Synching

- Accessing Wearable Data Layer
- Syncing Data Items
- Transferring Assets
- Sending and Receiving Messages
- Handling Data Layer Events

Creating Watch Faces

- Designing Watch Faces
- Watch Face Design Considerations
- Building a Watch Face Service
- Drawing Watch Faces
- Showing Information in Faces
- Creating Interactive Watch Faces

Location aware Android wear

- Location on Wear Device
- Getting Location on Android Wear
- Detecting On-Board GPS

- Handling GPS Location Disconnection Events
- Synchronizing Location Data with Handheld

Apps

- Stopwatch
- Simple Calculator App
- Location based Notification App
- Android Wear activity monitor app

Project

- Media Player App
- Weather App

Prerequisites:

- Basics and fundamentals of java
- Prior programming experience will be helpful.
- No previous knowledge of Kotlin and Android is required.

Who Should Attend:

- Anyone who wants to be an app developer: This is a complete course, just like my Complete Web, iOS and Apple Watch courses. It will teach you how to make money from your apps as well as how to code.
- Anyone who wants to learn to code: Java is a fantastic language to learn how to code with.
- Anyone who wants to understand how computers work: Learning to code is so much more than being able to make apps - knowing how computers work is your key to a hugely powerful world.
- It is recommended that you have a good understanding of Java, or take the Java Essentials.

Number of Hours: 40hrs

Key Features:

- One to One Training

- Online Training
- Fastrack & Normal Track
- Resume Modification
- Mock Interviews
- Video Tutorials
- Materials
- Real Time Projects
- Virtual Live Experience
- Preparing for Certification