ଓଡ଼ିଶା ରାଜ୍ୟ ମୁକ୍ତ ବିଶ୍ୱବିଦ୍ୟାଳୟ, ସମ୍ବଲପୁର, ଓଡ଼ିଶା
Odisha State Open University, Sambalpur, Odisha
Established by an Act of Government of Odisha.

DIPLOMA IN ANDROID APPLICATION
(DAP)

CSP-80

Fundamentals of Android Programming

Block

2

Android Development Environment
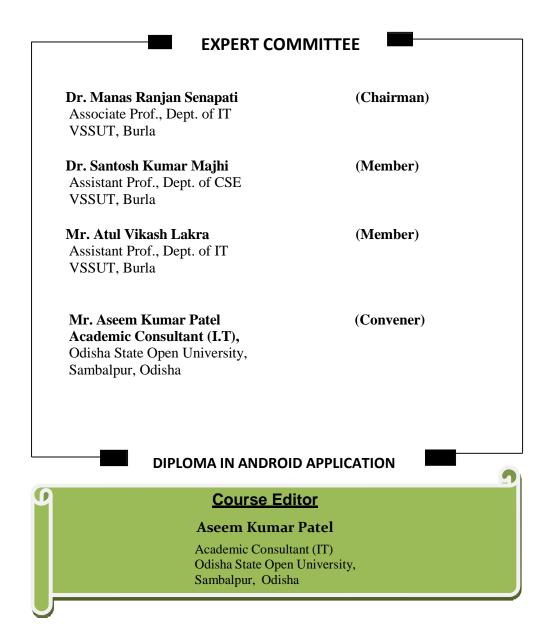
Unit -1
Android Development Platform

Unit -2

Installation and configuration of ADE

Unit -3

First Android programming

ଓଡ଼ିଶା ରାଜ୍ୟ ମୁକ୍ତ ବିଶ୍ୱବିଦ୍ୟାଳୟ, ସମ୍ବଲପୁର, ଓଡ଼ିଶା
Odisha State Open University, Sambalpur, Odisha
Established by an Act of Government of Odisha.

EXPERT COMMITTEE

Dr. Manas Ranjan Senapati   **(Chairman)**
Associate Prof., Dept. of IT
VSSUT, Burla

Dr. Santosh Kumar Majhi   **(Member)**
Assistant Prof., Dept. of CSE
VSSUT, Burla

Mr. Atul Vikash Lakra   **(Member)**
Assistant Prof., Dept. of IT
VSSUT, Burla

Mr. Aseem Kumar Patel   **(Convener)**
Academic Consultant (I.T),
Odisha State Open University,
Sambalpur, Odisha

DIPLOMA IN ANDROID APPLICATION

Course Editor

Aseem Kumar Patel

Academic Consultant (IT)
Odisha State Open University,
Sambalpur, Odisha

About this Course Material for OPEN DISTANCE LEARNING
Welcome to the course on "Fundamentals of Android Programming".
The book '**Introduction to Android**' has been produced by The Open University of Sri Lanka.
The course material titled "**Android Programming**" has been produced by GRAPHIC ERA HILL UNIVERSITY, Dehradun, India as a part of the OER for Skills Development initiative of Commonwealth of Learning (COL). All the study material produced by Odisha State Open University is structured in the same way, as outlined below.

UNIT-01

Learning Objective

After Compleation of this Unit we will learn about:

1) What is ADE Environment?
2) Know about Android development Platforms.
3) The background, and platform versions, system features, Android tools for the development environment
4) Reasons for Android Development
5) System Requirements for Android Development

Unit Structure

Unit 1

Android Development Platform

Introduction

In this unit you will be able to get familiar with available Android development platforms. You need to watch the video and install the required tools to start developing your first Android application.

Upon completion of this unit you should be able to:

Outcomes

▪ explain development platforms and distinguish each against their features and capabilities.

▪ describe the background, and platform versions, system features, Android tools for the development environment.

▪ configure the Android development environment in a computer.

Terminology

command line:	commands entered as inputs without IDE
SDK:	Software Development Kit
JDK:	Java Development Kit
Android Studio:	Official Android platform to develop Android apps

1.1 Reasons for Android Development

Today, mobile telephones have fundamentally changed the way of people interact. It is evident that mobile applications will be the future of handheld devices, Television and Automobile. Moreover, developers have started embedding Android in home appliances and other devices.

There are many reasons for the popularity of Android apps, such as:

- Google provides one window solution, as Play Store, to upload and download the application either free or with

minimal charges. For uploading and distributing the app, developers have no need of any approval of someone.

- Developer is the owner of his / her app and has the total control on product. However, Google has rights to unpublish any Android application in play store, if it is not complying with Google's licenses. For instance, if application contains malicious code or violate license, Google has right to unpublish the application.

- Android has open source operating system, open source software development kit (SDK) and good documentation.

- Android applications are not limited to mobile devices (Phones & Tabs), but can be run on TVs, wearable devices, vehicles and even refrigerators.

- (Source: First Thrust Towards Android, Android Programming, Course Material for Open Distance Learning, Commonwealth of Learning 2016)

1.2 Android Development Platforms, Features and Tools

In unit 2, you have learned Android architecture and major components of the Android platform. Let's look at the Android platform and the features they are providing.

Android Studio is the official IDE for Android development, and with a single download it includes everything you need to begin developing Android apps as you can see below

- IntelliJ IDE + Android Studio plugin

- Android SDK Tools

- Android Platform-tools

- A version of the Android platform

- Android Emulator with an Android system image including Google Play Services

Android Studio provides tools for building apps on every type of Android device. Code editing, debugging, performance tooling, a flexible build system, and an instant build or deploy system are included in Android studio. Let's see what are the systems requirements to install Android studio in different operating systems.

System Requirements

System requirements for Windows, Mac OS and Linux are given below.

Windows - Microsoft® Windows® 7/8/10 (32- or 64-bit)

- 3 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator

- 2 GB of available disk space minimum,

- 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- For accelerated emulator: 64-bit operating system and Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality

Mac - Mac® OS X® 10.10 (Yosemite) or higher, up to 10.12 (macOS Sierra)

- 3 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum,
- 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

Linux - GNOME or KDE desktop

- Tested on Ubuntu® 12.04, Precise Pangolin (64-bit distribution capable of running 32-bit applications)
- 64-bit distribution capable of running 32-bit applications
- GNU C Library (glibc) 2.19 or later
- 3 GB RAM minimum, 8 GB RAM recommended; plus 1 GB for the Android Emulator
- 2 GB of available disk space minimum,
- 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution

For accelerated emulator: Intel® processor with support for Intel® VT-x, Intel® EM64T (Intel® 64), and Execute Disable (XD) Bit functionality, or AMD processor with support for AMD Virtualization™ (AMD-V™).

(Source: https://source.android.com/source/requirements.html, CC:BY: 2.5)

Command Line Tools

The Android SDK tools available from the SDK Manager provide additional command-line tools to help you during your Android development. The tools are classified into two groups: SDK tools and platform tools. SDK tools are platform independent and are required no matter which Android platform you are developing on. Platform tools are

customized to support the features of the latest Android platform.

Additional Command Line Tools

The Android SDK tools available from the SDK Manager provide additional command-line tools to help you during your Android development. The tools are classified into two groups: SDK tools and platform tools. SDK tools are platform independent and are required no matter which Android platform you are developing on. Platform tools are customized to support the features of the latest Android platform.

SDK Tools

The SDK tools are installed with the SDK starter package and are periodically updated. The SDK tools are required if you are developing Android applications. The most important SDK tools include the Android SDK Manager (Android sdk), the AVD Manager (Android AVD) the emulator (emulator), and the Dalvik Debug Monitor Server (DDMS). A short summary of some frequently-used SDK tools is provided below.

Virtual Device Tools

a) Android Virtual Device Manager

The AVD Manager provides a graphical user interface in which you can create and manage Android Virtual Devices (AVDs) that run in the Android Emulator.

b) Android Emulator (emulator)

A QEMU(short for quick emulator) based device emulation tool that you can use to debug and test your applications in an actual Android run-time environment.

c) mksdcard

Helps you create a disk image that you can use with the emulator, to simulate the presence of an external storage card (such as an SD card).

Development Tools

Hierarchy Viewer (hierarchyviewer) - Provides a visual representation of the layout's View hierarchy with performance information for each node in the layout, and a magnified view of the display to closely examine the pixels in your layout.

SDK Manager

SDK Manager lets you manage SDK packages, such as installed platforms and system images.

sqlite3 - Lets you access the SQLite data files created and used by Android applications.

Debugging Tools

The debugging tools are further explained in the later units of this material.

a) Android Monitor

Android Monitor is integrated into Android Studio and provides logcat, memory, CPU, GPU, and network monitors for app debugging and analysis.

b) adb

Android Debug Bridge (adb) is a versatile command line tool that lets you communicate with an emulator instance or connected Android-powered device. It also provides access to the device shell.

c) Dalvik Debug Monitor Server (DDMS)

DDMS lets you debug Android apps.

d) Device Monitor

Android Device Monitor is a stand-alone tool that provides a graphical user interface for several Android application debugging and analysis tools.

e) Systrace

This tool lets you analyze the execution of your application in the context of system processes, to help diagnose display and performance issues.

f) traceview

Provides a graphical viewer for execution logs saved by your application.

g) Tracer for OpenGL ES

Allows you to capture OpenGL ES(the standard for Embedded Accelerated 3D Graphics) commands and frame-by-frame images to help you understand how your app is executing graphics commands.

Build Tools

a) apksigner

Signs APKs and checks whether APK signatures will be verified successfully on all platform versions that a given APK supports.

b) JOBB

Allows you to build encrypted and unencrypted APK expansion files in Opaque Binary Blob (OBB) format.

c) ProGuard

Shrinks, optimizes, and obfuscates your code by removing unused code and renaming classes, fields, and methods with semantically obscure names.

d) zipalign

Optimizes APK files by ensuring that all uncompressed data starts with a particular alignment relative to the start of the file.

Image Tools

a) Draw 9-patch

Allows you to easily create a NinePatch (class permits drawing a bitmap in nine or more sections) graphic using a WYSIWYG (What You See Is What You Get) editor. It also previews stretched versions of the image, and highlights the area in which content is allowed.

b) Etc1tool

A command line utility that lets you encode PNG images to the ETC1 compression standard and decode ETC1 compressed images back to PNG.

Platform Tools

The platform tools are typically updated every time you install a new SDK platform. Each update of the platform tools is backward compatible with older platforms. Usually, you directly use only one of the platform tools—the Android Debug Bridge (adb). Android Debug Bridge is a versatile tool that lets you manage the state of an emulator instance or Android-powered device. You can also use it to install an Android application (APK) file on a device.

The other platform tools, such as aidl, aapt, dexdump, and dx, are typically called by the Android build tools, so you rarely need to invoke these tools directly. As a general rule, you should rely on the build tools to call them as needed.

Note: The Android SDK provides additional shell tools that can be accessed through adb, such as bmgr and logcat.

a) bmgr

A shell tool you can use to interact with the Backup Manager on Android devices supporting API Level 8 or greater.

b) logcat

Provides a mechanism for collecting and viewing system debug output.

(Source: https://developer.android.com/studio/command-line/index.html CC:BY: 2.5)

Now, you have and learnt the systems requirements (hardware/software features) to set up the Android development platform and the Android command line tools. It is vital to determine the specific features of each Android version and how it has been developed to performing better. In unit 1 we learnt the history of Android and how each version of Android evolved. Next, we will summarize the Android platform versions with their unique features.

Android platform versions and specific features

There are rapid developments and new versions for the Android and Table 4.1 summarize the specific features of different Android platform versions up to now.

Table 4.1 summary the specific features of different Android platform versions

Android Platform version	Specific Features
Android 1.6 - Donut	Quick search box Screen size diversity Android market
Android 2.1 - Eclair	Google maps navigation Home screen customization Speech-to- Text
Android 2.2- Froyo	Voice actions Portable Hotspot Performance
Android 2.3- Gingerbread	Gaming APIs Near Field Communication (NFC) Battery Management
Android 3.0- Honeycomb	Tablet-friendly design System bar Quick settings

Android 4.0- Ice cream Sandwich	Custom home screen Data usage control Android Beam
Android 4.1- Jelly Bean	Google now Actionable Notifications Account switching
Android 4.4- Kitkat	Voice: OK Google Immersive Design Smart Dialer
Android 5.0- Lollipop	Material Design Multiscreen Notifications
Android 6.0- Marshmallow	Now on tap Permissions Battery works smarter
Android 7.0 - Nougat	Multi Locale language settings Multi-window view Quick switch between apps Data Saver Notification Controls Display Size

You can read more information online about the relative numbers of devices that are running different versions in the following link.

https://developer.android.com/about/dashboards/index.html

Thus, creating apps in Android for various mobile devices are increasing day by day. Since developing native apps is expensive, the demand for cross platform app development tools is also increasing. It is essential to know cross platforms and tools for mobile application development to develop apps for enhancing the market capacity.

You can watch the online presentation given in the link below to get familiar with the cross platforms for mobile application development.
bit.ly/XPlatformMobileDev

1.3 Python for Android Mobile App Development

Python is a high-level programming language that is widely used in web development, app development, analyzing and computing scientific and numeric data, creating desktop GUIs, and for software development.

Python is the most taught programming language at the school and college level for the fact that it has several applications in real life.

Python is a powerful high-level language that can be used to create android and desktop apps from scratch. Just to give you a hint of how powerful this language is, Dropbox is created in Python.

Python is used in a wide variety of application domains as it can easily be connected with C, Objective-C, Java, or FORTRAN. It runs on all major operating systems, like Windows, Linux/Unix, OS/2, Mac, Amiga, etc. With the help of Python, we can create any type of mobile applications, like Calibre, OpenStack, Ubuntu Software Center, World of Tanks, Quora, BitTorrent, Reddit, Spotify, YouTube, Instagram, and many more.

1.4 Why Python for Mobile App Development?

1. Python runs on all major operating systems such as Windows, Linux/Unix, OS/2, Mac, Amiga, etc.

2. The language provide constructs intended to enable clear programs on both a small and a large scale.

3. Python provides more tools for both the developers and the system administrators.

4. Many companies like GOOGLE, Yahoo!, and IBM use this Python, because it's a fun and a dynamic language.

5. It provides faster development and portability allows for the same application to run across platforms.

6. Python is packed with rich libraries and many add-on packages to tackle specific tasks.

The features of Python is natural expression of procedural code, strong introspection capabilities, very clear, readable syntax, intuitive object orientation, very high level dynamic data types, extensions and modules easily written in C, C++ (or Java for

Jython, or .NET languages for IronPython), extensive standard libraries and third party modules for virtually every task, full modularity, supporting hierarchical packages, exception-based error handling and embeddable within applications as a scripting interface. Python is widely used in Web application development. It is also used in a wide range of non-scripting contexts. Python plays well with COM, .NET, CORBA, Java and it often compared to Tcl, Perl, Ruby, Scheme or Java.

1.5 Role of Python in Mobile Applications?

There are a variety of applications in various fields including business, entertainment, utilities, hospitality sector, games and much more, these apps are made sure to fit to various screen sizes be it iPad, iPhone, iPod Touch, laptop, palmtop etc. Mobile Application Development has become a leading business sector due to its increasing scope. For every person in the world, it is quite impossible to live without mobile. It plays an important role in all life. There are various software systems used by the designers for the mobile phones such as Symbian, J2Me, Android, Flash light, Python, Lazarus, BREW, etc.

This is a nice cross platform python framework which works for Android, Win7, Linux, Mac. It is a great tool for writing both simple scripts and complex, multi-threaded applications. The great thing about having Python on Android is the opportunity to use the untold thousands of lines of code already written and freely available. iPhone or Android App Developers take a number of cross platform development techniques in order to provide a great mobile application to the customers. The top tools used for cross-formatting mobile application development are RhoMobile, PhoneGap, Appcelerator, Sencha Touch, MoSync, Whoop, WidgetPad, GENWI, AppMakr, Mippin, SwebApps, MobiCart, etc.

Appcelerator – This platform uses web technologies and develops great apps for desktop, mobile and tablet applications. The languages which are used to develop apps are HTML, Python, PHP and few others.

MoSync SDK – It is used to build mobile applications iOS, Android, Windows Phone 7, Symbian, Windows Mobile, javaME and Blackberry. It uses common programming languages such as PHP, JavaScript, Ruby, & Python for cross platform Mobile Application Development.

1.6 Mobile Application written in Python

1. **Aarlogic C05/3 –** Ready to use GSM /GPS tracking PCB with Python development on board with support of test server based on Google Maps.

2. **AppBackup –** an app for jailbroken iOS devices that lets one back-up and restore settings and data from App Store apps.

3. **Pyroute –** a GPS-capable mapping/routing application for mobile devices.

Android Google provides Android Scripting Environment (ASE) which allows scripting languages (Python included) to run on Android. The excellent features of Python allows it to plays an important role in mobile applications.

Activity

Activity 1.1

Explore most popular cross platforms and name three major cross platforms.

Briefly describe one of the major cross platform with the features and uses of them.

Hint: Check your answers with Answer guide at the end of this unit.

Now you are aware of the native and cross platform for mobile application development. We will now explore how the development environment is set up and configured.

Android.

Unit summary

The first step to start on Android based application development is to set up the development environment. Therefore, it is essential to know how to configure development environment and installing tools for Android application development.

In this unit, you have learned Android development platforms, tools and cross platforms. At the last section of the unit, you learnt about setting up the Android development environment by installing the latest JDK and Android Studio. It is also important to keep the Android studio IDE and Android SDK tools up to date at the development environment.