# Possible Questions for 1st Tutorial

# (with a sample)

# 1. Android - Environment Setup

## Questions

* 1. Write two main sophisticated technologies for developing Android application and what are the requirements (hardware and software) to set-up Android studio.

There are so many sophisticated Technologies are available to develop android applications, the familiar technologies, which are predominantly using tools as follows

• Android Studio

• Eclipse IDE (Deprecated)

**Step 1 - System Requirements**

We can start our Android application development on either of the following operating systems −

• Microsoft® Windows® 10/8/7/Vista/2003 (32 or 64-bit)

• Mac® OS X® 10.8.5 or higher, up to 10.9 (Mavericks)

• GNOME or KDE desktop

All the required tools to develop Android applications are open source and can be downloaded from the Web. Following is the list of software's we will need before we start our Android application programming.

• Java JDK5 or later version

• Java Runtime Environment (JRE) 6

• Android Studio

**Step 2 - Setup Android Studio**

Android Studio is the official IDE(integrated development environment) for android application development.It works based on IntelliJ IDEA, we can download the latest version of android studio from Android Studio 2.2 Download. After download then install it.

**Step 3 – Create Android Virtual Device(AVD)**

To test our Android applications, we will need a virtual Android device. So before we start writing our code, let us create an Android virtual device. Launch Android AVD Manager Clicking AVD\_Manager icon.

* 1. How to set-up Java Development Kit (JDK) for Android?

We can download the latest version of Java JDK from Oracle's Java site − Java SE Downloads. We will find instructions for installing JDK in downloaded files, follow the given instructions to install and configure the setup. Finally set PATH and JAVA\_HOME environment variables to refer to the directory that contains java and javac, typically java\_install\_dir/bin and java\_install\_dir respectively.

If we are running Windows and installed the JDK in C:\jdk1.8.0\_102, we would have to put the following line in our C:\autoexec.bat file.

set PATH=C:\jdk1.8.0\_102\bin;%PATH%

set JAVA\_HOME=C:\jdk1.8.0\_102

Alternatively, we could also right-click on My Computer, select Properties, then Advanced, then Environment Variables. Then, we would update the PATH value and press the OK button.

On Linux, if the SDK is installed in /usr/local/jdk1.8.0\_102 and we use the C shell, we would put the following code into our .cshrc file.

setenv PATH /usr/local/jdk1.8.0\_102/bin:$PATH

setenv JAVA\_HOME /usr/local/jdk1.8.0\_102

Alternatively, if we use Android studio, then it will know automatically where we have installed our Java.

* 1. How to set-up Android SDK (Software Development Kit) for Android??

We can download the latest version of Android SDK from Android official website − Android SDK Downloads. If we are installing SDK on Windows machine, then we will find a installer\_rXX-windows.exe, so just download and run this exe which will launch Android SDK Tool Set up wizard to guide we throughout of the installation, so just follow the instructions carefully. Finally we will have Android SDK Tools installed on our machine. If we are installing SDK either on Mac OS or Linux, check the instructions provided along with the downloaded android-sdk\_rXX-macosx.zip file for Mac OS and android-sdk\_rXX-linux.tgz file for Linux.

* 1. How to set-up Android Development Tools (ADT) Plug-in for Android??

This step will help us in setting Android Development Tool plug-in for Eclipse. Let's start with launching Eclipse and then, choose Help > Software Updates > Install New Software. This will display a dialogue box. Now use Add button to add ADT Plug-in as name and https://dl-ssl.google.com/android/eclipse/ as the location. Then click OK to add this location, as soon as we will click OK button to add this location, Eclipse starts searching for the plug-in available the given location and finally lists down the found plug-ins. Now select all the listed plug-ins using Select All button and click Next button which will guide we ahead to install Android Development Tools and other required plug-ins.

* 1. How to set-up Eclipse IDE (Integrated Development Environment) for Android?

To install Eclipse IDE, download the latest Eclipse binaries from https://www.eclipse.org/downloads/. Once we downloaded the installation, unpack the binary distribution into a convenient location. For example in C:\eclipse on windows, or /usr/local/eclipse on Linux and finally set PATH variable appropriately.Eclipse can be started by executing the following commands on windows machine, or we can simply double click on eclipse.exe

%C:\eclipse\eclipse.exe

Eclipse can be started by executing the following commands on Linux machine −

$/usr/local/eclipse/eclipse

[Minimum requirements (hardware and software)]

**2. Android –Platform, Feature, Version etc.**

**Questions**

1. What is Mobile Platform ? Write some examples of good platform. 2

Mobile platforms are basically those that allow software and services to be run on devices. Examples of mobile platforms include Palm, BlackBerry, iPhone, Android and Windows Mobile.

1. What is WAP? Write some features of WAP.3

Wireless Application Protocol (WAP) is a technical standard for accessing information over a mobile wireless network.

Some of the key features of WAP technology are:

1. A programming model similar to the Internet
2. Wireless Markup Language (WML)
3. WML Script
4. Wireless Telephony Application Interface (WTAI)
5. Optimized protocol stack
6. What is Android and why? Write some applications of Android.2-3
7. What’s are Android API levels? Make a list of Android version name sequentially with API levels.4-5

API Levels generally mean that as a programmer, you can communicate with the devices' built in functions and functionality. As the API level increases, functionality adds up (although some of it can get deprecated).

|  |  |  |
| --- | --- | --- |
| **Platform Version** | **API Level** | **VERSION\_CODE** |
| Android 8.0 | 26-27 | OREO |
| Android 7.0 | 24-25 | NOUGAT |
| Android 6.0 | 23 | MARSHMALLOW |
| Android 5.1 | 22 | LOLLIPOP\_MR1 |
| Android 5.0 | 21 | LOLLIPOP |
| Android 4.4W | 20 | KITKAT\_WATCH |
| Android 4.4 | 19 | KITKAT |
| Android 4.3 | 18 | JELLY\_BEAN\_MR2 |
| Android 4.2, 4.2.2 | 17 | JELLY\_BEAN\_MR1 |
| Android 4.1, 4.1.1 | 16 | JELLY\_BEAN |
| Android 4.0.3, 4.0.4 | 15 | ICE\_CREAM\_SANDWICH\_MR1 |
| Android 4.0, 4.0.1, 4.0.2 | 14 | ICE\_CREAM\_SANDWICH |
| Android 3.2 | 13 | HONEYCOMB\_MR2 |
| Android 3.1.x | 12 | HONEYCOMB\_MR1 |
| Android 3.0.x | 11 | HONEYCOMB |
| Android 2.3.4  Android 2.3.3 | 10 | GINGERBREAD\_MR1 |
| Android 2.3.2  Android 2.3.1  Android 2.3 | 9 | GINGERBREAD |
| Android 2.2.x | 8 | FROYO |
| Android 2.1.x | 7 | ECLAIR\_MR1 |
| Android 2.0.1 | 6 | ECLAIR\_0\_1 |
| Android 2.0 | 5 | ÉCLAIR |
| Android 1.6 | 4 | DONUT |
| Android 1.5 | 3 | CUPCAKE |
| Android 1.1 | 2 | BASE\_1\_1 |
| Android 1.0 | 1 | BASE |

1. Explain Eclipse IDE and Android Development Tools (ADT) Plug-in with their purpose in Android. 4-5

Android offers a custom plugin for the Eclipse IDE, called Android Development Tools (ADT). This plugin provides a powerful, integrated environment in which to develop Android apps.

It extends the capabilities of Eclipse to let you quickly set up new Android projects, build an app UI, debug your app, and export signed (or unsigned) app packages (APKs) for distribution.

Developing in Eclipse with ADT is highly recommended and is the fastest way to get started. With the guided project setup it provides, as well as tools integration, custom XML editors, and debug output pane, ADT gives you an incredible boost in developing Android applications.

1. What are the possible next Android version names? 1-2

Android 9.0 Popcorn (by Google)

Android 9.0 Pastry

Android 9.0 Pasta

Android 9.0 Popover

Android 9.0 PanCake

Android 9.0 Pumpkin Pie

Android Pudding

Android Pie

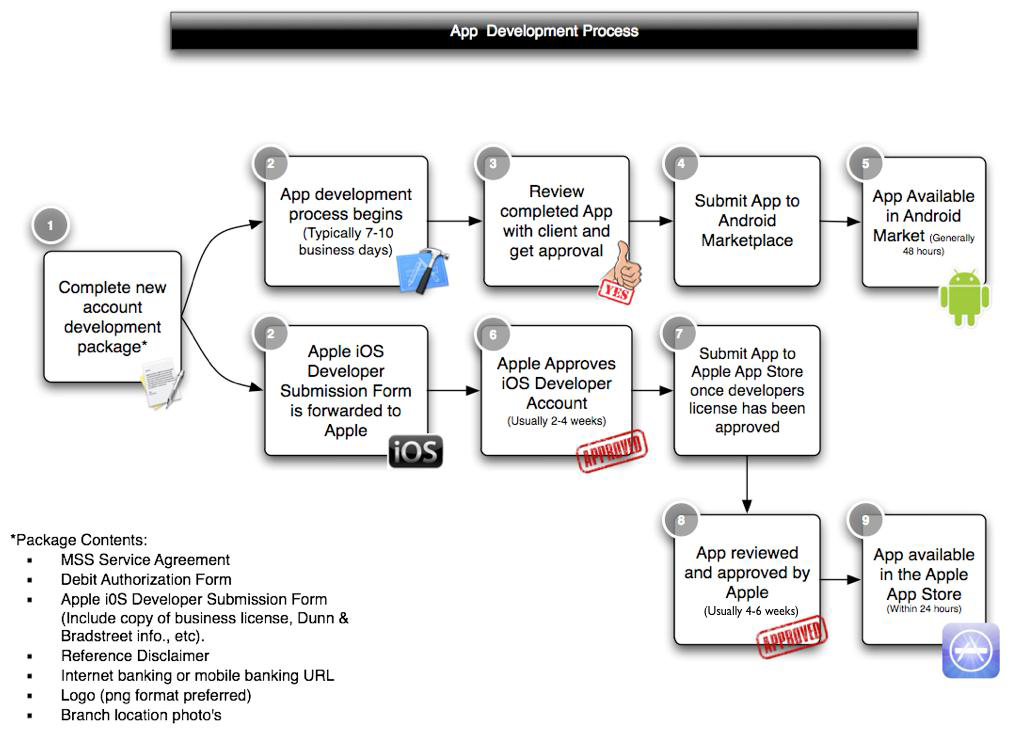
Android Pista

Android Pilu

Android 9.0 Petha

Android 9.0 Phirni

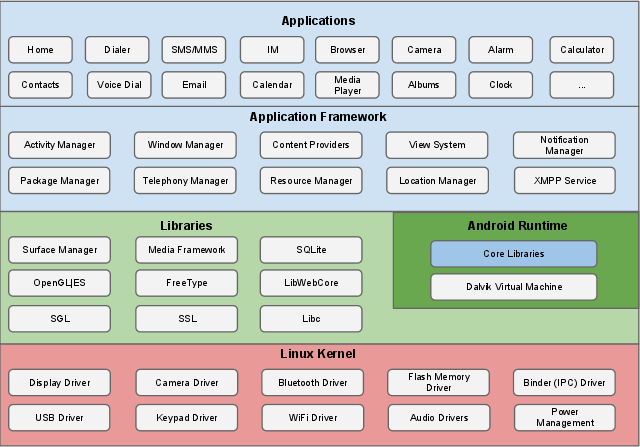
1. Sketch the Block Diagram of Android & IOS Application Development.2
2. Sketch the Block Diagram of Android & IOS Development.2



**3. Android – Architecture & Components**

**Question**

1. Draw the Architecture of Android with All the Components.



1. Write Down the Definition & Function of:
2. Linux Karnel

**Linux Kernel:** The Linux Kernel handles core system services and acts as a hardware abstraction layer (HAL) between the physical hardware of the device and the Android SDK. Some of the major functionalities include low level memory management, process management, networking and other OS related services.

1. Android Runtime

**Android Runtime:** This includes the Dalvik Virtual Machine (DVM) and the core java libraries. Based on java VM, the Dalvik design has been optimized for the low memory requirements of mobile devices. It allows multiple VMs to run concurrently.

1. Java Virtual Machine

JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed. JVMs are available for many hardware and software platforms.

The JVM performs following operation:

Loads code

Verifies code

Executes code

Provides runtime environment

1. Delvik Virtual Machine

The Dalvik Virtual Machine (DVM) is an android virtual machine optimized for mobile devices. It optimizes the virtual machine for memory, battery life and performance.

Dalvik is a name of a town in Iceland. The Dalvik VM was written by Dan Bornstein.

The Dex compiler converts the class files into the .dex file that run on the Dalvik VM. Multiple class files are converted into one dex file.

1. Android Libraries

The Android native libraries are all shared libraries written in C or C++, compiled for the particular hardware architecture used by the phone. Some of the important libraries include OpenGL for 2D and 3D graphics, SQLite database and the WebKit library for browsing HTML content.

1. Application Components

Application components are the essential building blocks of an Android application. These components are loosely coupled by the application manifest file AndroidManifest.xml that describes each component of the application and how they interact.

There are following four main components that can be used within an Android application –

Activities

Services

Broadcast Receivers

Content providers

1. Write Down the Function of Following Android Libraries:
2. Android.app
3. Android.content
4. Android.database
5. Android.opengl
6. Androd.os
7. Android.view
8. Android.widget
9. Android.webkit.
10. Write Short Notes on the Following Application Framework Key Services:
11. Activity Manager
12. Content Providers
13. Resource Manager
14. Notification Manager
15. View System
16. Write Short Notes on Following Application Components:
17. Activities
18. Services
19. Broadcast Receivers
20. Content Providers
21. Fragments

A Fragment represents a behavior or a portion of user interface in a FragmentActivity.

Android Fragment is the part of activity, it is also known as sub-activity. There can be more than one fragment in an activity. Fragments represent multiple screen inside one activity.

Android fragment lifecycle is affected by activity lifecycle because fragments are included in activity.

Each fragment has its own life cycle methods that is affected by activity life cycle because fragments are embedded in activity.

1. Views

Usually View is used as arguments in methods which act as some kind of listener. For example when we have more than 1 Button in our layout and we set onClick on them, we create a method like this:

public void onClick(View view){

}

1. Layouts

A layout defines the structure for a user interface in your app, such as in an activity. All elements in the layout are built using a hierarchy of View and ViewGroup objects.

There are number of Layouts provided by Android which you will use in almost all the Android applications to provide different view, look and feel.

1 Linear Layout

LinearLayout is a view group that aligns all children in a single direction, vertically or horizontally.

2 Relative Layout

RelativeLayout is a view group that displays child views in relative positions.

3 Table Layout

TableLayout is a view that groups views into rows and columns.

4 Absolute Layout

AbsoluteLayout enables you to specify the exact location of its children.

5 Frame Layout

The FrameLayout is a placeholder on screen that you can use to display a single view.

6 List View

ListView is a view group that displays a list of scrollable items.

7 Grid View

GridView is a ViewGroup that displays items in a two-dimensional, scrollable grid.

1. Intents

An Android Intent is an object carrying intent, i.e. a message from one component to another component either inside or outside of the application. Intents can communicate messages among any of the three core components of an application -- Activities, Services, and BroadcastReceivers.

1. Resources

Resources are the additional files and static content that your code uses, such as bitmaps, layout definitions, user interface strings, animation instructions, and more.

Resource files are stored in folders located in the res folder, including:

drawable: For images and icons

layout: For layout resource files

menu: For menu items

mipmap: For pre-calculated, optimized collections of app icons used by the Launcher

values: For colors, dimensions, strings, and styles (theme attributes)

1. Manifest

Every app project must have an AndroidManifest.xml file (with precisely that name) at the root of the project source set. The manifest file describes essential information about your app to the Android build tools, the Android operating system, and Google Play.

**Login**

1. How to create a simple Login Screen project Using Android Studio?

**Android Frontend (XML UI)**

1. Define Android View Group & View.

The Android View class and ViewGroup class are two very central classes in Android apps.

The **View** class is a superclass for all GUI components in Android. For instance, the TextView class which is used to display text labels in Android apps is a subclass of View. Android contains the following commonly used View subclasses:

TextView

EditText

ImageView

ProgressBar

Button

ImageButton

CheckBox

The **ViewGroup** class is a subclass of the View class. ViewGroup instances work as containers for View instances to group View instances together. Android contains the following commonly used ViewGroup subclasses:

LinearLayout

RelativeLayout

ListView

GridView

1. Write Short Notes on Different Android Layout Types.

The six different layouts are

1. Linear Layout

2. Relative Layout

3. Table Layout

4. Grid layout

5. Tab Layout

6. List View

**Linear Layout:** A Layout that arranges its children in a single column or a single row.

**Relative Layout:** Relative Layout is a view group that displays child views in relative positions.

**Table Layout**: A layout that arranges its children into rows and columns. A TableLayout consists of a number of TableRow objects, each defining a row.

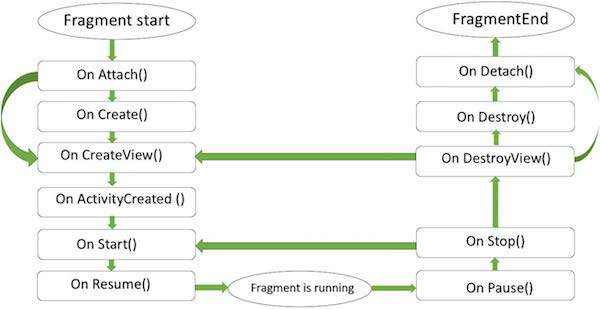
**Grid layout:** A layout that places its children in a rectangular grid.

**Tab Layout:** TabLayout provides a horizontal layout to display tabs.

**List View:** A view that shows items in a vertically scrolling list

1. Differentiate Among Linear Layout, Relative Layout & Absolute Layout.
2. What are the Differences between Frame Layout & Table Layout?
3. What is Android Fragment? Write Down the Block Diagram of Fragment Lifecycle.

A Fragment represents a behavior or a portion of user interface in a FragmentActivity.



1. How to Use Fragments? Describe Different Types of Android Fragments.

A Fragment represents a behavior or a portion of user interface in a FragmentActivity. You can combine multiple fragments in a single activity to build a multi-pane UI and reuse a fragment in multiple activities.

Basically fragments are divided as three stages as shown below.

Single frame fragments − Single frame fragments are using for hand hold devices like mobiles, here we can show only one fragment as a view.

List fragments − fragments having special list view is called as list fragment

Fragments transaction − Using with fragment transaction. we can move one fragment to another fragment.

**Institute of Information Technology (IIT)**

Jahangirnagar University, Savar Dhaka

Tutorial –1

IT 4227: Mobile application Developments

Time-45 Min. Answer any **Two** Questions Marks: 20

**All parts of a particular question must be answered consecutively**

|  |  |  |
| --- | --- | --- |
| 1(a) | What is Android and why it is popular platform? Mention few features of Android operating system | 3 |
| (b) | What is API ? Mention the android version name sequentially. | 3 |
| (c) | How many layers are in Android architecture? Mention the android sections name with few components. | 4 |
|  |  |  |
| 2(a) | Explain AndroidManifest.xml file in detail.  Every application must have an AndroidManifest.xml file (with precisely that name) in its root directory. The manifest presents essential information about the application to the Android system, information the system must have before it can run any of the application's code.  the manifest does the following:   * It names the Java package for the application. The package name serves as a unique identifier for the application. * It describes the components of the application — the activities, services, broadcast receivers, and content providers that the application is composed of. It names the classes that implement each of the components and publishes their capabilities (for example, which [Intent](https://stuff.mit.edu/afs/sipb/project/android/docs/reference/android/content/Intent.html) messages they can handle). These declarations let the Android system know what the components are and under what conditions they can be launched. * It determines which processes will host application components. * It declares which permissions the application must have in order to access protected parts of the API and interact with other applications. * It also declares the permissions that others are required to have in order to interact with the application's components. * It lists the [Instrumentation](https://stuff.mit.edu/afs/sipb/project/android/docs/reference/android/app/Instrumentation.html) classes that provide profiling and other information as the application is running. These declarations are present in the manifest only while the application is being developed and tested; they're removed before the application is published. * It declares the minimum level of the Android API that the application requires. * It lists the libraries that the application must be linked against. | 3 |
| (b) | Draw Android activities interface and labeling it. | 3 |
| (c) | Differentiate the following terms:   1. wrap\_content and match\_parent 2. Java and JavaScript. 3. LinearLayout and RelativeLayout 4. JDK and IDEs | 4 |
|  |  |  |
| 3(a) | What is orientation in Android? | 2 |
| (b) | Sketch the android application development process | 3 |
| (c) | Interpret the following android code line by line. | 5 |
|  | <?xml version="1.0" encoding="utf-8"?>  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent"    <TextView android:id="@+id/text"  android:textSize="50sp"  android:textColor="#0832ca"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hello, I am a TextView" />  <Button android:id="@+id/button"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Hello, I am a Button" />    </LinearLayout> |  |