# 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.

[**Android Studio**](https://developer.android.com/studio/index.html)– The official IDE, based on the community-created IntelliJIDEA (see below).

[**Eclipse**](https://eclipse.org/downloads/)– Before Android Studio, this was the official Android development environment. Used to code Java but can be expanded to other languages via plugins, it is still a powerful tool.

* 1. How to set-up Java Development Kit (JDK) for Android?
  2. How to set-up Android SDK (Software Development Kit) for Android??
  3. How to set-up Android Development Tools (ADT) Plug-in for Android??
  4. How to set-up Eclipse IDE (Integrated Development Environment) for Android?

[Minimum requirements (hardware and software)]

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

**Questions**

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

**A mobile application platform is a suite of software tools used for designing, creating and maintaining mobile applications.**

**Example mobile platforms are Windows Phone, BlackBerry, Symbian and iPhone OS.**

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

**[WAP is] the worldwide standard for providing Internet communications and advanced telephony services on digital mobile phones, pagers, personal digital assistants, and other wireless terminals.**

**WAP stands for Wireless Application Protocol.**

**Features:**

Though WAP is a new technology, but it reuse the concepts found on the Internet

**WML:** You must be using HTML language to develop your web-based application.

**WMLSCRIPT:** you must be using Java Script or VB script to enhance the functionality of your web applications.

**Wireless Telephony Application Interface (WTAI):**

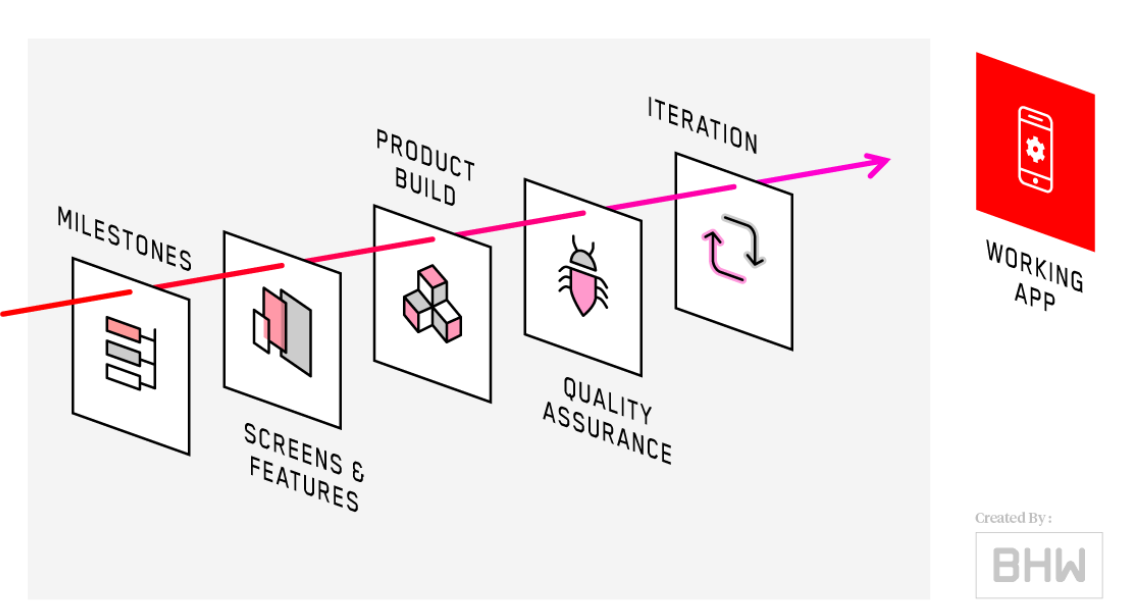
1. What is Android and why? Write some applications of Android.2-3
2. What’s are Android API levels? Make a list of Android version name sequentially with API levels.4-5
3. Explain Eclipse IDE and Android Development Tools (ADT) Plug-in with their purpose in Android. 4-5

**Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE.[6] It contains a base workspace and an extensible plug-in system for customizing the environment.**

**Android Development Tools**

**Android Development Tools (ADT) is a discontinued Google-**provided plug-in for the Eclipse IDE that is designed to provide an integrated environment in which to build Android applications. ADT extends the abilities of Eclipse to let developers set up new Android projects, create an application UI, add packages based on the Android Framework API, debug their applications using the Android SDK tools, and export signed (or unsigned) .apk files in order to distribute their applications. It is freely available to download. It was the official IDE for Android but was replaced by Android Studio (based on IntelliJ IDEA Community Edition).[49] ADT is officially deprecated since the end of 2015, and now Google is focused on Android Studio as the official Android IDE.[50] The Android Device Monitor that shipped with ADT was built on the Eclipse Platform. This tool still ships with Android Studio.

1. What are the possible next Android version names? 1-2
2. Sketch the Block Diagram of Android & IOS Application Development.2

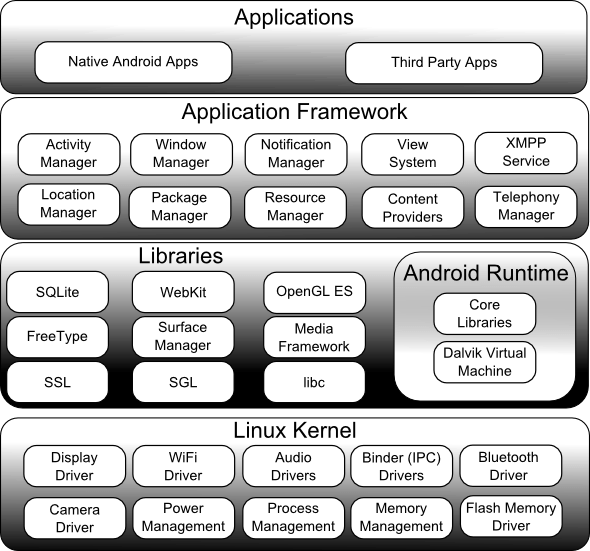


1. Sketch the Block Diagram of Android & IOS Development.2

**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
3. Android Runtime
4. Java Virtual Machine
5. Delvik Virtual Machine
6. Android Libraries
7. Application Components
8. Write Down the Function of Following Android Libraries:
9. Android.app
10. Android.content
11. Android.database
12. Android.opengl
13. Androd.os
14. Android.view
15. Android.widget
16. Android.webkit.
17. Write Short Notes on the Following Application Framework Key Services:
18. Activity Manager

**This class gives information about, and interacts with, activities, services, and the containing process.**

**A number of the methods in this class are for debugging or informational purposes and they should not be used to affect any runtime behavior of your app. These methods are called out as such in the method level documentation.**

1. Content Providers

[Calendar Provider](https://developer.android.com/guide/topics/providers/calendar-provider.html)

A content provider manages access to a central repository of data. A provider is part of an Android application, which often provides its own UI for working with the data

1. Resource Manager

If by resource manager you mean applications like Clean master or CCleaner, they help clear out waste Memory and storage which is basically freeing up RAM so that your phone runs better.. Some also clean cache which is a part storing temporary files created by applications.

1. Notification Manager

**Class to notify the user of events that happen. This is how you tell the user that something has happened in the background.**

Notifications can take different forms:

* A persistent icon that goes in the status bar and is accessible through the launcher, (when the user selects it, a designated Intent can be launched),
* Turning on or flashing LEDs on the device, or
* Alerting the user by flashing the backlight, playing a sound, or vibrating.

1. View System

This class represents the basic building block for user interface components. A View occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for *widgets*, which are used to create interactive UI components (buttons, text fields, etc.). The [ViewGroup](https://developer.android.com/reference/android/view/ViewGroup.html) subclass is the base class for *layouts*, which are invisible containers that hold other Views (or other ViewGroups) and define their layout properties.

1. Write Short Notes on Following Application Components:
2. Activities
3. Services
4. Broadcast Receivers
5. Content Providers
6. Fragments
7. Views
8. Layouts
9. Intents
10. Resources
11. Manifest

**Login**

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

**Android Frontend (XML UI)**

1. Define Android View Group & View.

## View

1. View objects are the basic building blocks of User Interface(UI) elements in Android.
2. View is a simple rectangle box which responds to the user's actions.
3. Examples are EditText, Button, CheckBox etc..
4. View refers to the android.view.View class, which is the base class of all UI classes.

## ViewGroup

1. ViewGroup is the invisible container. It holds View and ViewGroup
2. For example, LinearLayout is the ViewGroup that contains Button(View), and other Layouts also.
3. ViewGroup is the base class for Layouts.
4. Write Short Notes on Different Android Layout Types.

|  |  |
| --- | --- |
| 1 | [**Linear Layout**](https://www.tutorialspoint.com/android/android_linear_layout.htm)  LinearLayout is a view group that aligns all children in a single direction, vertically or horizontally. |
| 2 | [**Relative Layout**](https://www.tutorialspoint.com/android/android_relative_layout.htm)  RelativeLayout is a view group that displays child views in relative positions. |
| 3 | [**Table Layout**](https://www.tutorialspoint.com/android/android_table_layout.htm)  TableLayout is a view that groups views into rows and columns. |
| 4 | [**Absolute Layout**](https://www.tutorialspoint.com/android/android_absolute_layout.htm)  AbsoluteLayout enables you to specify the exact location of its children. |
| 5 | [**Frame Layout**](https://www.tutorialspoint.com/android/android_frame_layout.htm)  The FrameLayout is a placeholder on screen that you can use to display a single view. |
| 6 | [**List View**](https://www.tutorialspoint.com/android/android_list_view.htm)  ListView is a view group that displays a list of scrollable items. |
| 7 | [**Grid View**](https://www.tutorialspoint.com/android/android_grid_view.htm)  GridView is a ViewGroup that displays items in a two-dimensional, scrollable grid. |

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**is a piece of an activity which enable more modular activity design. It will not be wrong if we say, a fragment is a kind of **sub-activity**.

First of all you need to understand what are lifecycle methods are and when are they called/invoked. Lifecycle methods are basically invoked at the different state of your Activty/Fragment. For example when you first launch your activity the following flow of events/methods are called depending upon the state of your activity. For example : When your activity is first launched OnCreate is called, when your activity is no longer visible then onStop is called. So basically you first need to learn at which state are these different activities called.Below is a great referential flowchart for the same.



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

There are four types of fragments:

* ListFragment
* DialogFragment
* PreferenceFragment
* WebViewFragment

## ListFragment

This fragment is similar to ListActivity and contains a ListView view by default. It is used for displaying a list of items. In our previous sample code, we used ListFragment; see the Creating and managing fragments section for ListFragment.

## DialogFragment

This fragment displays a dialog on top of its owner activity. In the following sample application, we are going to create a fragment that has a **Delete**button. When the button is clicked, a **DialogFragment** dialog box will be displayed. ...

**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. | 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> |  |

1.

The **AndroidManifest.xml file** *contains information of your package*, including components of the application such as activities, services, broadcast receivers, content providers etc.

It performs some other tasks also:

* It is **responsible to protect the application** to access any protected parts by providing the permissions.
* It also **declares the android api** that the application is going to use.
* It **lists the instrumentation classes**. The instrumentation classes provides profiling and other informations. These informations are removed just before the application is published etc.
* This is the required xml file for all the android application and located inside the root directory.

2.

WRAP\_CONTENT, which means that the view wants to be just big enough to enclose its content (plus padding)-- Special value for the height or width requested by a View. The view should be only big enough to enclose its content (plus padding).

[MATCH\_PARENT](https://developer.android.com/reference/android/view/ViewGroup.LayoutParams.html#MATCH_PARENT)

Special value for the height or width requested by a View. The view should be as big as its parent (minus padding). Introduced in API Level 8.

**FILL PARENT:** Setting a top level layout or control to fill\_parent will force it to take up the whole screen.

**WRAP\_CONTENT,** which means that the View wants to be just big enough to enclose its content (plus padding)

Key **differences between Java and JavaScript**:**Java** is an OOP programming language while **Java**Script is an OOP scripting language. **Java** creates applications that run in a virtual machine or browser while **JavaScript** code is run on a browser only. **Java**code needs to be compiled while **JavaScript** code are all in text.

LinearLayout means you can align views one by one (vertically/ horizontally).

RelativeLayout means based on relation of views from its parents and other views.

JDK, or Java Development Kit, is a bunch of compiled components that aid you in developing Java programs. a JDK contains the functions, objects, and other components for you to use in order to develop your own software.

- An IDE, or Integrated Development Environment, is a software tool that acts as the communication platform between you, the programmer, and the JDK. It's the tool through which you can "talk" to the JDK, and use it to write your Java programs.

3.

The **screenOrientation** is the attribute of activity element. The orientation of android activity can be portrait, landscape, sensor, unspecified etc. You need to define it in the AndroidManifest.xml file. For example:

The common values for screenOrientation attribute are as follows:

|  |  |
| --- | --- |
| **Value** | **Description** |
| unspecified | It is the default value. In such case, system chooses the orientation. |
| portrait | taller not wider |
| landscape | wider not taller |
| sensor | orientation is determined by the device orientation sensor. |

**Mobile application development process:**

