Assignment 1

You are free to search online for answers to these questions. But do answer (type) them by yourself.

It is a good practice to search for the correct answer by yourself.

It is recommended you follow the docs https://developer.android.com/guide and search for terms related to android development here

Q 1. What is Android Studio and its alternatives? (answer in not more than 2 sentences)

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA.

There are two ways you can develop android applications other than using Android Studio.

1. Eclipse

Eclipse is an open source IDE that lets you work in a wide variety of platforms. Android SDK can be merged with Eclipse using the ADT(Android Developer Tools) package that can be downloaded from Google itself.

2. Hybrid Application Development

This is the lightest way of compiling and running Android applications. It take much lesser time compared to both Studio and Eclipse. You have to code using HTML5 and Javascript, which is then compiled with some hybrid app engine like Cordova or Sencha and then converted into an APK file for running on the device.

Q 2. What (programming) languages are supported by Android Studio? (list and give 1 sentence brief for each supported language)

The languages you might consider learning for Android development include:

Java – Java is the official language of Android development and is supported by Android Studio. It has a steep learning curve however.

Kotlin – Kotlin was recently introduced as a secondary “official” Java language. It is similar to Java in many ways but is a little easier to get your head around.

C/C++ — Android Studio also supports C++ with the use of the Java NDK. This allows for native coding applications, which can be handy for things like games. C++ is more complicated though.

C# — C# is a slightly more beginner-friendly alternative to C or C++ that obfuscates more code. It’s supported by some very handy tools like Unity and Xamarin which are great for game development and for cross-platform development.

BASIC – A bonus option is to learn BASIC and try the B4A IDE from Anywhere Software. This is an easy but powerful tool, though definitely much more niche!

Corona/LUA – Another cross-platform tool build on LUA. It massively simplifies the app-building process and allows you to call native libraries.

PhoneGap (HTML, CSS, JavaScript) – If you already know how to build interactive web pages, then you can use this knowledge with PhoneGap to build a more basic cross-platform app.

Q 3. Give example for following -

1) **Application Name** - pdf_viewer

2) **Package Name** - com.androidmaker.pdf_viewer.

3) **Application ID** - com.androidmaker.pdf_viewer.

Package Name and Application ID are Same By Default. But Package name can be changed, Not Application ID (Since its Unique)

Rules for Naming: -

It must have at least two segments (one or more dots).

Each segment must start with a letter.

All characters must be alphanumeric or an underscore [a-zA-Z0-9_]

Q 4. Is it possible to change application ID or package name for an application? (Answer in max 4 sentences)

YES,Its Possible to Change Package Name,But Not the Application ID,Since its Unique For Different App Published.

If you change the application ID, Google Play Store treats the APK as a completely different app. So once you publish your app, you should never change the application ID.

Q 5. Briefly explain what is Android-Sdk. (1-2 sentences of explanation)

A software development kit that enables developers to create applications for the Android platform. The Android SDK includes sample projects with source code, development tools, an emulator, and required libraries to build Android applications.

Q 6. What is the purpose of AndroidManifest.xml file?

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 provide profiling and other informations. These information's 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

Q 7. What is the purpose of gradle in context of Android Studio or Android development. (short answer, max 4 sentences)

Gradle is a build system (open source) which is used to automate building, testing, deployment etc. "Build. gradle" are scripts where one can automate the tasks. For example, the simple task to copy some files from one directory to another can be performed by Gradle build script before the actual build process happens.

Q 8. Explain resources in terms of Android development. Give example. (1 sentence explanation and example)

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

Example :-

```
MyProject/
   src/
      MyActivity.java
   res/
      drawable/
         graphic.png
      layout/
         main.xml
         info.xml
      mipmap/
         icon.png
      values/
         strings.xml
```

Q 9. Explain following in context of Android development-

1) Activity

An activity is the entry point for interacting with the user. It represents a single screen with a user interface. For example, an email app might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. Although the activities work together to form a cohesive user experience in the email app, each one is independent of the others. As such, a different app can start any one of these activities if the email app allows it. For example, a camera app can start the activity in the email app that composes new mail to allow the user to share a picture.

2) Launcher activity

Displays a list of all activities which can be performed for a given intent. Launches when clicked.

3) Events

Events are a useful way to collect data about a user's interaction with interactive components of Applications. Like button presses or screen touch etc. The Android framework maintains an event queue as first-in, first-out (FIFO) basis. You can capture these events in your program and take appropriate action as per requirements.

There are following three concepts related to Android Event Management −

Event Listeners – Defined Below

Event Listeners Registration **–** Event Registration is the process by which an Event Handler gets registered with an Event Listener so that the handler is called when the Event Listener fires the event.

Event Handlers − When an event happens and we have registered an event listener for the event, the event listener calls the Event Handlers, which is the method that actually handles the event.

4) Event listeners

− An event listener is an interface in the View class that contains a single callback method. These methods will be called by the Android framework when the View to which the listener has been registered is triggered by user interaction with the item in the UI**.**

5) Views

A View occupies a rectangular area on the screen and is responsible for drawing and event handling. The View class is a superclass for all GUI components in Android. Commonly used Views are: EditText.

Q 10. Give example of views available in Android. (list upto 10 and minimum of 5 different views)

Here we have some of the most commonly used android View classes:

TextView.

EditText.

Button.

ImageView.

ImageButton.

CheckBox.

RadioButton.

ListView.

GridView

DatePicker

Spinner