TYBBA(CA)-SEM-VI

604- Android Programming

**2 Marks**

**Answer the following questions**

**1.       What are the features of Android?**

Ans :

1. Application Framework: It enables reuse and replacements of components.

2. Dalvik Virtual Machine: It is optimized for mobile device.

3. Integrated Browser: It is based on open source web kit engine.

4. Handset layouts: Optimized Graphics: It is powered by a custom 2D graphics Library 3D Graphics based on the OpenGL ES 1.0 Specification.

5. Storage: SQLite, a lightweight relational database.

6. Media Support: Android supports the following audio/video/still media formats: H.263, H.264 (in 3GP or MP4 container), MPEG 4 SP, AMR, AMR WB (in 3GP container), AAC, HE AAC (in MP4 or 3GP container), MP3, MIDI, Ogg Vorbis,  WAV, JPEG, PNG, GIF, BMP

7. Connectivity: GSM/EDGE, IDEN, CDMA, EVDO, UMTS, Bluetooth, WiFi.

8. Messaging: SMS and MMS are available forms of messaging, also support  Android Cloud to Device Messaging Framework (C2DM).

9. Multiple Language Support: Multiple Languages Available in Android.

10. Additional hardware support: Android can use video/still cameras, touch screens, GPS, accelerometers, gyroscopes, magnetometers.

**2.       Explain the Android stack diagrammatically/ Android architecture.**

Linux kernel: This is the kernel on which Android is based. This layer contains all the low level device drivers for the various hardware components of an Android device.

Libraries: These contain all the code that provides the main features of an Android OS. For example, the SQLite library provides database support so that an application can use it for data storage. The Webkit library provides functionalities for web browsing.

Android runtime: At the same layer as the libraries, the Android provides a set of core libraries that enable developers to write Android apps using the Java programming language.

Application framework: Exposes the various capabilities of the Android OS to application developers so that they can make use of them in their applications.

(\***REFER DIAGRAM FROM BOOK)**

**3.       Explain SDK platform in deatil.**

The Android SDK (Software Development Kit) is the most important software of android which is installed. The Android SDK provides to test android applications, the API libraries, an emulator, documentation, sample code, developer tools and tutorials which help you to build, test and debug apps of Android SDK contains Debugger  Libraries Emulator Documentation Sample Code Tutorials Android SDK is made up of two main parts: the tools and the packages. When you first install the SDK, all you obtain are the base tools. Is helps you to develop applications. The packages are the records specific to a particular version of Android.

**4.       What is activity?**

An activity is a window that contains the user interface of your applications. An application can have zero or more activities. Typically, applications have one or more activities, and the main aim of an activity is to interact with the user. From the moment an activity appears on the screen to the moment it is hidden, it goes through a number of stages, known as an activity's life cycle. Understanding the life cycle of an activity is vital to ensuring that your application works correctly.

**5.       What is intent?**

The Android operating system uses an asynchronous messaging mechanism to match task requests with the appropriate Activity. Each request is packaged as an Intent The intent generally consists of activities to be done, the parameter over which this activity is to be performed, and the application to perform this action. Examples of processes defined as intents:

1. Start of certain activities.

2. Displaying a web page.

3. Answering a call.

4. Dialing a number.

**6.       With the help of diagram describe activity life cycle**

Method        Description

onCreate      called when activity is first created.

onStart        called when activity is becoming visible to the user.

onResume   called when activity will start interacting with the user.

onPause       called when activity is not visible to the user.

onStop         called when activity is no longer visible to the user.

onRestart     called after your activity is stopped, prior to start.

onDestroy    called before the activity is destroyed.

**(\*Refer Diagram from book)**

**7.       Describe lifecycle of fragment diagrammatically.**

In Android, the fragment is the part of Activity which represents a portion of User Interface(UI) on the screen. It is the modular section of the android activity that is very helpful in creating UI designs that are flexible in nature and auto-adjustable based on the device screen size. The UI flexibility on all devices improves the user experience and adaptability of the application. Fragments can exist only inside an activity as its lifecycle is dependent on the lifecycle of host activity. For example, if the host activity is paused, then all the methods and operations of the fragment related to that activity will stop functioning, thus fragment is also termed as sub-activity.

**(\*Refer Diagram from book)**

**8.       Enlist type of intents.**

Explicit Intent − It going to connect the internal world of an application such as start activity or send data between two activities.

Implicit Intents − It going to connect with out side application such as call, mail, phone,see any website ..etc.

**9.       How to calling built-in applications using intents? Explain using example**

Ans :

Calling built in applications using intents in Android is important as it give user more choices. For example in an app you want user make a call or open a document (pdf) let you use the in-build apps, take off the burden in developing those apps.

Android intent example

Lets get started with how to use intent in android. The following Try It Out demonstrates how to call some of the built-in applications commonly found on an Android device.

Creating New Project

1. First step : Open Android Studio and Create a new project by selecting 'Start a new Android Studio Project'. Give it a name MyIntent and a package name in.ruks. You can have your own application package names. Click next, choose minimum SDK, in ours case it is API 14. Android Studio will create the the new Android Project with all the necessary files.

Set up Project Layout

We are going to use 6 built-in application in this example. They are -

1. Gallery 2. Call Log 3. Contact 4. Dailler 5. Browser 6. Map

2. Second step : Now the time to set up the layout of our project. We are going to call 6 built-in applications using intents, hence we will place a button for each in activity\_main.xml. You can drag and drop 6 button on the layout or simply create by coding the activity\_main.xml file.

3. Third step : It's bad idea to have the hard coded string in the project. So in this android intent example code, we are given name to each of the button within the strings.xml, which is under the values folder in Android Studio. Add the code below in it -

Creating event handler

4. Fourth step : In our MainActivity.java file, we are going to declare the button for each of the Button we have created in our layout file.

5. Fifth step : Our Buttons are ready to have some action, now its time to attach the event handler to them. One by one we are going to call setOnClickListener method on each of our Button and call built-in applications.

Declare Activity in AndroidManifest

6. Final step : If you have basic knowledge of developing android apps then you must know that, every activity you have in your application must be declared in your AndroidManifest.xml file.

**10.     What is UI?**

The user interface (UI) for an Android app is built as a hierarchy of layouts and widgets. The layouts are ViewGroup objects, containers that control how their child views are positioned on the screen. Widgets are View objects, UI components such as buttons and text boxes.

**11.     What is layout? Enlist Layout Managers.**

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. A View usually draws something the user can see and interact with. Whereas a ViewGroup is an invisible container that defines the layout structure for View and other ViewGroup objects

layout Manager

1. A LinearLayout lines up its views in one row or column.

2. A FrameLayout is a simple layout manager used to display one view.

3. A RelativeLayout is a layout manager in which the views are positioned in relation to each other or to the parent.

**12.     What is view?**

View is a basic building block of UI (User Interface) in android. A view is a small rectangular box that responds to user inputs. Eg: EditText, Button, CheckBox, etc. ViewGroup is an invisible container of other views (child views) and other ViewGroup.

**13.     What is Action bar?**

The app bar, also known as the action bar, is one of the most important design elements in your app's activities, because it provides a visual structure and interactive elements that are familiar to users.

**14.     Describe the following LayoutManagers with example:**

a.       LinearLayout

Linear layout is a simple layout used in android for layout designing. In the Linear layout all the elements are displayed in linear fashion means all the childs/elements of a linear layout are displayed according to its orientation. The value for orientation property can be either horizontal or vertical.

There are two types of linear layout orientation:

1. Vertical

2. Horizontal

b.       Tablelayout

A layout that arranges its children into rows and columns. A TableLayout consists of a number of TableRow objects, each defining a row (actually, you can have other children, which will be explained below). TableLayout containers do not display border lines for their rows, columns, or cells. Each row has zero or more cells; each cell can hold one View object. The table has as many columns as the row with the most cells. A table can leave cells empty. Cells can span columns, as they can in HTML.

**15.     Explain the term utilizing action bar in detail.**

Action Bar is one of the important parts of any application, whether it’s a web application or a mobile app. Today we will learn how to implement an action bar in android apps using the ActionBar component.

Android ActionBar is a menu bar that runs across the top of the activity screen in android. Android ActionBar can contain menu items which become visible when the user clicks the “menu” button.

In general an ActionBar consists of the following four components:

App Icon: App branding logo or icon will be displayed here

View Control: A dedicated space to display Application title. Also provides option to switch between views by adding spinner or tabbed navigation

Action Buttons: Some important actions of the app can be added here

Action Overflow: All unimportant action will be shown as a menu

**16.     What is ScrollView? Explain with example.**

a ScrollView is a view group that is used to make vertically scrollable views. A scroll view contains a single direct child only. In order to place multiple views in the scroll view, one needs to make a view group(like LinearLayout) as a direct child and then we can define many views inside it. A ScrollView supports Vertical scrolling only, so in order to create a horizontally scrollable view, HorizontalScrollView is used.

**17.     With the help of diagram describe FrameLayout**

Frame Layout is one of the simplest layout to organize view controls. They are designed to block an area on the screen. Frame Layout should be used to hold child view, because it can be difficult to display single views at a specific area on the screen without overlapping each other.

We can add multiple children to a FrameLayout and control their position by assigning gravity to each child, using the android:layout\_gravity attribute.

**(\*Refer Diagram from book)**

**18.     What is the TextView?**

A TextView displays text to the user and optionally allows them to edit it. A TextView is a complete text editor, however the basic class is configured to not allow editing.

**19.     With the help of example explain RadioButton.**

RadioButton is a two states button which is either checked or unchecked. If a single radio button is unchecked, we can click it to make checked radio button. Once a radio button is checked, it cannot be marked as unchecked by user.

RadioButton is generally used with RadioGroup. RadioGroup contains several radio buttons, marking one radio button as checked makes all other radio buttons as unchecked.

**20.     What is meant by button? Describe with example.**

In Android applications, a Button is a user interface that is used to perform some action when clicked or tapped. It is a very common widget in Android and developers often use it.

**21.     What is the EditText?**

A EditText is an overlay over TextView that configures itself to be editable. It is the predefined subclass of TextView that includes rich editing capabilities.

**22.     Explain the AutoCompleteTextView in detail.**

A  AutoCompleteTextView is a view that is similar to EditText, except that it shows a list of completion suggestions automatically while the user is typing.

The AutoCompleteTextView is a type of edit text in android which gives suggestions to the user if the user types something in the AutoCompleteTextView. This type of edit text we can see while we register on some websites. If we type “In” it will suggest India, Indonesia, West Indies ….. etc. Like this, the AutoCompleteTextView works. Let us see the implementation of AutoCompleteTextView in XML and Java . Here XML is used to create the layout and java code is used to implement the main function of AutoCompleteTextView .

**23.     Write short note on: ProgressBar.**

A progress bar is a graphical control element used to visualize the progression of an extended computer operation, such as a download, file transfer, or installation. Sometimes, the graphic is accompanied by a textual representation of the progress in a percent format.

**24.     What is ToggleButton? How to create it? Explain with example.**

Android Toggle Button can be used to display checked/unchecked (On/Off) state on the button.

It is beneficial if user have to change the setting between two states. It can be used to On/Off Sound, Wifi, Bluetooth etc.

Since Android 4.0, there is another type of toggle button called switch that provides slider control.

Android ToggleButton and Switch both are the subclasses of CompoundButton class.

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context="example.javatpoint.com.togglebutton.MainActivity">

      <ToggleButton

        android:id="@+id/toggleButton"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_marginLeft="8dp"

        android:layout\_marginTop="80dp"

        android:text="ToggleButton"

        android:textOff="Off"

        android:textOn="On"

        app:layout\_constraintEnd\_toStartOf="@+id/toggleButton2"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent" />

  </android.support.constraint.ConstraintLayout>

**25.     Describe the following terms:**

a.       DatePicker : Android Date Picker allows you to select the date consisting of day, month and year in your custom user interface. For this functionality android provides DatePicker and DatePickerDialog components.

b.       TimePicker : Android Time Picker allows you to select the time of day in either 24 hour or AM/PM mode. The time consists of hours, minutes and clock format. Android provides this functionality through TimePicker class.  
**26.     What is the gallery in Android?**

Gallery is an offline photo gallery. You can use it to edit, find, and manage photos without being connected to the Internet. Gallery also works with SD cards on Android devices. Tip: Gallery is available on Android devices.

**27.     What is GridView in Android?**

A GridView is a type of AdapterView that displays items in a two-dimensional scrolling grid. Items are inserted into this grid layout from a database or from an array. The adapter is used for displaying this data, setAdapter() method is used to join the adapter with GridView. The main function of the adapter in GridView is to fetch data from a database or array and insert each piece of data in an appropriate item that will be displayed in GridView.

**28.     What is ImageView in Android?**

ImageView class is used to display any kind of image resource in the android application either it can be android.graphics.Bitmap or android.graphics.drawable.Drawable (it is a general abstraction for anything that can be drawn in Android). ImageView class or android.widget.ImageView inherits the android.view.View class which is the subclass of Kotlin.Any class. Application of ImageView is also in applying tints to an image in order to reuse a drawable resource and create overlays on background images.

**29.     Write short note on: Creating the helper methods.**

1. Provides common methods which are required by multiple classes in the project.

2. Helper methods are generally public and static so that these can be invoked independently.

3. Each methods of a helper class should work independent of other methods of same class.

**30.     What is menu? Enlist its types.**

The traditional menu definition is a list of food or drink items available for purchase, or a list of food or drink items that will be served. That can be a seasonal menu, a fixed menu, or something in between. While the majority of the time the meaning of menu is applied to food and beverages, that’s not always the case.

The word menu is also used frequently in the context of electronic devices and computer programs. There it usually refers to a navigable list of options a user can interact with.

The five types of menus most commonly used are 1. a la carte menus, 2. static menus, 3. du jour menus, 3. cycle menus, and 3. fixed menus.

**31.     What is thread?**

A thread is a thread of execution in a program. The Java Virtual Machine allows an application to have multiple threads of execution running concurrently. Every thread has a priority. Threads with higher priority are executed in preference to threads with lower priority.

**32.     Enlist Types of handlers.**

Runnable, long) , postDelayed(Runnable, Object, long) , sendEmptyMessage(int) , sendMessage(Message) , sendMessageAtTime(Message, long) , and sendMessageDelayed(Message, long)

**33.     What is SMS?**

We’re all familiar with SMS, or standard text messages. After all, it’s one of the oldest and most commonly used methods of mobile communication. While SMS is seemingly humdrum in the modern age, there’s a surprising amount of coordination and technology working in the background to send such seemingly simple messages. So let’s take a look at how it all works.

**35.     What is Broadcast Receiver?**

A broadcast receiver (receiver) is an Android component which allows you to register for system or application events. All registered receivers for an event are notified by the Android runtime once this event happens.

**36.     What is meant by database?**

SQLite Database is an open-source database provided in Android which is used to store data inside the user's device in the form of a Text file. We can perform so many operations on this data such as adding new data, updating, reading, and deleting this data.

**37.     What is SQLite?**

SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL database engine. SQLite is the most used database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day.

**38.     Enlist feature of SQLite.**

1. Transactions are atomic, consistent, isolated, and durable (ACID) even after system crashes and power failures.

2. Zero-configuration - no setup or administration needed.

3. Full-featured SQL implementation with advanced capabilities like partial indexes, indexes on expressions, JSON, common table expressions, and window functions. (Omitted features)

4. A complete database is stored in a single cross-platform disk file. Great for use as an application file format.

5. Supports terabyte-sized databases and gigabyte-sized strings and blobs. (See limits.html.)

6. Small code footprint: less than 750KiB fully configured or much less with optional features omitted.

7. Simple, easy to use API.

8. Fast: In some cases, SQLite is faster than direct filesystem I/O

9. Written in ANSI-C. TCL bindings included. Bindings for dozens of other languages available separately.

10. Well-commented source code with 100% branch test coverage.

11. Available as a single ANSI-C source-code file that is easy to compile and hence is easy to add into a larger project.

12. Self-contained: no external dependencies.

**39.     How to close database using SQLite?**

close( conn ) closes the SQLite connection by using the MATLAB® interface to SQLite. The SQLite connection object remains open until you close it using the close function. Always close this object when you finish using it.

**40.     What is meant by Google map?**

Google Maps is a Web-based service that provides detailed information about geographical regions and sites around the world. In addition to conventional road maps, Google Maps offers aerial and satellite views of many places. In some cities, Google Maps offers street views comprising photographs taken from vehicles.

**41.     How to create Google map?**

1. On your computer, sign in to My Maps.

2. Click Create a new map.

3. Go to the top left and click "Untitled map."

4. Give your map a name and description.

**42.     What is marker? How to add markers in Google map.**

Markers are objects of type Marker , and are added to the map with the GoogleMap. addMarker(markerOptions) method. Markers are designed to be interactive. They receive click events by default, and are often used with event listeners to bring up info windows.

How to Add :

1. On your computer, sign in to My Maps.

2. Open or create a map. A map can have up to 10,000 lines, shapes, or places.

3. Click Add marker .

4. Select a layer and click where to put the place. A layer can have 2,000 lines, shapes, or places.

5. Give your place a name.

6. Click Save

# 3 Marks

**Answer the following questions**

1. **Android**

Android is a software package and linux based operating system for mobile devices such as tablet computers and smartphones.

It is developed by Google and later the OHA (Open Handset Alliance). Java language is mainly used to write the android code even though other languages can be used.

The goal of android project is to create a successful real-world product that improves the mobile experience for end users.

There are many code names of android such as Lollipop, Kitkat, Jelly Bean, Ice cream Sandwich, Froyo, Ecliar, Donut etc which is covered in next page.

1. **JDK**

The Java Development Kit (JDK) is a cross-platformed software development environment that offers a collection of tools and libraries necessary for developing Java-based software applications and applets. It is a core package used in Java, along with the JVM (Java Virtual Machine) and the JRE (Java Runtime Environment).

Beginners often get confused with JRE and JDK, if you are only interested in running Java programs on your machine then you can easily do it using Java Runtime Environment. However, if you would like to develop a Java-based software application then along with JRE you may need some additional necessary tools, which is called JDK.

**3.       Android runtime**

Dalvik virtual machine: The Android runtime also includes the Dalvik virtual machine, which enables every Android application to run in its own process with its own instance of the Dalvik virtual machine (Android applications are compiled into the Dalvik executables). Dalvik is a specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU.

Core Libraries: These are different from Java SE and Java ME libraries. But these libraries provide most of the functionalities defined in the Java SE libraries.

1. Data Structure 2. File Access 3. Network Access 4. Utilities 5. Graphics

**4.       Intent**

An intent is an abstract description of an operation to be performed. It can be used with startActivity to launch an Activity, broadcastIntent to send it to any interested BroadcastReceiver components, and Context.startService(Intent) or Context.bindService(Intent, ServiceConnection, int) to communicate with a background Service.

An Intent provides a facility for performing late runtime binding between the code in different applications. Its most significant use is in the launching of activities, where it can be thought of as the glue between activities. It is basically a passive data structure holding an abstract description of an action to be performed.

**5.       Toast**

A toast provides simple feedback about an operation in a small popup. It only fills the amount of space required for the message and the current activity remains visible and interactive. Toasts automatically disappear after a timeout. Use the makeText() method, which takes the following parameters:

The application Context.

The text that should appear to the user. The duration that the toast should remain on the screen. The makeText() method returns a properly initialized Toast object.

**6.       Thread**

A thread is a thread of execution in a program. The Java Virtual Machine allows an application to have multiple threads of execution running concurrently.

Every thread has a priority. Threads with higher priority are executed in preference to threads with lower priority. Each thread may or may not also be marked as a daemon. When code running in some thread creates a new Thread object, the new thread has its priority initially set equal to the priority of the creating thread, and is a daemon thread if and only if the creating thread is a daemon.

**7.       Runnable**

The Runnable interface should be implemented by any class whose instances are intended to be executed by a thread. The class must define a method of no arguments called run.

This interface is designed to provide a common protocol for objects that wish to execute code while they are active. For example, Runnable is implemented by class Thread. Being active simply means that a thread has been started and has not yet been stopped.

In addition, Runnable provides the means for a class to be active while not subclassing Thread. A class that implements Runnable can run without subclassing Thread by instantiating a Thread instance and passing itself in as the target.

**8.       Asynctask**

AsyncTask was intended to enable proper and easy use of the UI thread. However, the most common use case was for integrating into UI, and that would cause Context leaks, missed callbacks, or crashes on configuration changes. It also has inconsistent behavior on different versions of the platform, swallows exceptions from doInBackground, and does not provide much utility over using Executors directly.

AsyncTask is designed to be a helper class around Thread and Handler and does not constitute a generic threading framework. AsyncTasks should ideally be used for short operations (a few seconds at the most.) If you need to keep threads running for long periods of time, it is highly recommended you use the various APIs provided by the java.util.concurrent package such as Executor, ThreadPoolExecutor and FutureTask.

An asynchronous task is defined by a computation that runs on a background thread and whose result is published on the UI thread. An asynchronous task is defined by 3 generic types, called Params, Progress and Result, and 4 steps, called onPreExecute, doInBackground, onProgressUpdate and onPostExecute.

**9.       Bounded**

A bound service is the server in a client-server interface. It allows components (such as activities) to bind to the service, send requests, receive responses, and perform interprocess communication (IPC). A bound service typically lives only while it serves another application component and does not run in the background indefinitely.

This document describes how to create a bound service, including how to bind to the service from other application components. For additional information about services in general, such as how to deliver notifications from a service and set the service to run in the foreground, refer to the Services document.

**10.     SQLiteDatabase**

SQLite is a opensource SQL database that stores data to a text file on a device. Android comes in with built in SQLite database implementation. SQLite supports all the relational database features

Database – Package The main package is android.database.sqlite that contains the classes to manage your own databases

Database – Creation

In order to create a database you just need to call this method openOrCreateDatabase with your database name and mode as a parameter. It returns an instance of SQLite database which you have to receive in your own object.Its syntax is given below

SQLiteDatabase mydatabase = openOrCreateDatabase("your database name",MODE\_PRIVATE,null);

**11.     SQLiteOpenHelper**

A helper class to manage database creation and version management.

You create a subclass implementing onCreate(SQLiteDatabase), onUpgrade(SQLiteDatabase, int, int) and optionally onOpen(SQLiteDatabase), and this class takes care of opening the database if it exists, creating it if it does not, and upgrading it as necessary. Transactions are used to make sure the database is always in a sensible state.

This class makes it easy for ContentProvider implementations to defer opening and upgrading the database until first use, to avoid blocking application startup with long-running database upgrades.

For an example, see the NotePadProvider class in the NotePad sample application, in the samples/ directory of the SDK.

**12.     Geocoding and Reverse Geocoding.**

**GeoCoding** : A class for handling geocoding and reverse geocoding. Geocoding is the process of transforming a street address or other description of a location into a (latitude, longitude) coordinate. Reverse geocoding is the process of transforming a (latitude, longitude) coordinate into a (partial) address. The amount of detail in a reverse geocoded location description may vary, for example one might contain the full street address of the closest building, while another might contain only a city name and postal code. The Geocoder class requires a backend service that is not included in the core android framework. The Geocoder query methods will return an empty list if there no backend service in the platform. Use the isPresent() method to determine whether a Geocoder implementation exists.

**Rev. Geocoding** : Reverse-Geocoding is a process used to convert coordinates (latitude and longitude) to human-readable addresses. This is not exactly the opposite of Geocoding. In Geocoding, the Place is associated with a name and fixed coordinates. These coordinates are Double in nature. Negligible change in these coordinates may still refer to the same place, but we shall never get the place name as it is associated with only those fixed coordinates. Therefore, we shall definitely get the complete address in reverse geocoding, but the place name is not guaranteed. Through this article, we will show you an example of how to perform reverse-geocoding in Android.

# 4 Marks

**Answer the following questions**

**1.       Enlist reasons for why use Android OS?**

There are so many reasons you should choose the Android platform for mobile application development.

**1. Zero/negligible development cost:**

The development tools like Android SDK, JDK, and Eclipse IDE, etc. are free to download for the android mobile application development. Also, Google charges a small fee of $25, to distribute your mobile app on the Android Market.

**2. Open Source:**

The Android OS is an open-source platform based on the Linux kernel and multiple open-source libraries. In this way, developers are free to contribute to or extend the platform as necessary for building mobile apps which run on Android devices.

**3. Multi-Platform Support:**

In the market, there is a wide range of hardware devices powered by the Android OS, including many different phones and tablets. Even the development of android mobile apps can occur on Windows, Mac OS, or Linux.

**4. Multi-Carrier Support:**

World wide a large number of telecom carriers like Airtel, Vodafone, Idea Cellular, AT&T Mobility, BSNL, etc. are supporting Android-powered phones.

**5. Open Distribution Model:**

Android Market place (Google Play store) has very few restrictions on the content or functionality of an android app. So the developer can distribute their app through the Google Play store and as well other distribution channels like Amazon’s app store.

**2.       Write short note on: History of Android.**

The history and versions of android are interesting to know. The code names of android ranges from A to J currently, such as Aestro, Blender, Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwitch, Jelly Bean, KitKat and Lollipop. Let's understand the android history in a sequence.

1) Initially, Andy Rubin founded Android Incorporation in Palo Alto, California, United States in October, 2003. 2) In 17th August 2005, Google acquired android Incorporation. Since then, it is in the subsidiary of Google Incorporation.

3) The key employees of Android Incorporation are Andy Rubin, Rich Miner, Chris White and Nick Sears. 4) Originally intended for camera but shifted to smart phones later because of low market for camera only.

5) Android is the nick name of Andy Rubin given by coworkers because of his love to robots. 6) In 2007, Google announces the development of android OS.

7) In 2008, HTC launched the first android mobile.

**Version        Code name  API Level**

1.5      Cupcake      3

1.6     Donut 4

2.1      Eclair 7

2.2     Froyo 8

2.3     Gingerbread 9 and 10

3.1 and 3.3   Honeycomb 12 and 13

4.0     Ice Cream Sandwitch       15

4.1, 4.2 and 4.3      Jelly Bean    16, 17 and 18

4.4     KitKat         19

5.0     Lollipop       21

6.0     Marshmallow        23

7.0     Nougat        24-25

8.0     Oreo  26-27

**3.       What is meant by AVD?**

1. The Android Virtual Device Manager allows you to create Android Virtual Devices (AVDs), which you can then run to emulate a device on your computer. There's an important but subtle distinction between simulation and emulation.

2. Simulation means that the virtual device is merely a façade that simulates how an actual physical device might behave, but does not run the targeted operating system. The iOS development environment uses simulation, and this is probably a good choice for iOS given the limited number of devices available for that platform.

3. With emulation, however, your computer sets aside a block of memory to reproduce the environment found on the device that the emulator is emulating. Android Studio uses emulation, which means the Android Virtual Device Manager launches a sandboxed version of the Linux kernel and the entire Android stack in order to emulate the environment found on the physical Android device.

4. Although emulation provides a much more faithful environment on which to test your apps than simulation does, booting up an AVD can drag into the minutes, depending on the speed of your computer. The good news is that after your emulator is active in memory, it remains responsive. Nevertheless, if you have an Android phone or tablet, we recommend using the physical device to test your apps, rather than using an AVD.

**4.       Write short note on: Android Emulator.**

1. The Android emulator runs one full Android system stack, down to the kernel level, including a set of preinstalled applications (such as the dialer) that you can access from your applications.

2. We can select the version of the Android system we wish to run the emulator by configuring AVDs and we can also modify the mobile device skin and key mappings.

3. While launching the emulator at the runtime, we can use a variety of commands and options to control its behavior.

4. The Android emulator offers dynamic binary translation of device machine code to the OS and processor architecture of your development machine.

The android emulator supports many hardware features likely to be found on mobile devices, including:

1. AN ARMV5 CPU and the corresponding Memory Managements Unit (MMU).

2.A 16-bit LCD display.

3. One or more keyboards.

**5.       What is ListView? How to use it in Android application? Explain with example.**

Android ListView is a view which groups several items and display them in vertical scrollable list. The list items are automatically inserted to the list using an Adapter that pulls content from a source such as an array or database.

An adapter actually bridges between UI components and the data source that fill data into UI Component. Adapter holds the data and send the data to adapter view, the view can takes the data from adapter view and shows the data on different views like as spinner, list view, grid view etc.

The ListView and GridView are subclasses of AdapterView and they can be populated by binding them to an Adapter, which retrieves data from an external source and creates a View that represents each data entry.

Android provides several subclasses of Adapter that are useful for retrieving different kinds of data and building views for an AdapterView ( i.e. ListView or GridView). The common adapters are ArrayAdapter,Base Adapter, CursorAdapter, SimpleCursorAdapter,SpinnerAdapter and WrapperListAdapter. We will see separate examples for both the adapters.

**6.       Explain ListFragment with example.**

Static library support version of the framework's ListFragment. Used to write apps that run on platforms prior to Android 3.0. When running on Android 3.0 or above, this implementation is still used.

Step   Description

1        You will use Android Studio to create an Android application and name it as SimpleListFragment under a package com.example.tutorialspoint7.myapplication, with blank Activity.

2        Modify the string file, which has placed at res/values/string.xml to add new string constants

3        Create a layout called list\_fragment.xml under the directory res/layout to define your list fragments. and add fragment tag(<fragment>) to your activity\_main.xml

4        Create a myListFragment.java, which is placed at java/myListFragment.java and it contained onCreateView(),onActivityCreated() and OnItemClickListener()

5        Run the application to launch Android emulator and verify the result of the changes done in the application.

**7.       What is DialogFragment? Explain its uses.**

DialogFragment is a utility class of android development that is used to show a Dialog window, Floating on top of an activity window in an android application. This feature is added on API Level 11 and Deprecated on API Level 28. It has its own lifecycle which makes this class very useful for dialog box creation. Here is a sample video to understand what we are going to build in this article and what actually a Dialog Fragment is-

Methods of DialogFragment

Below methods are given to control the flow of DialogFragment, with the proper usage of these methods dialog boxes can be controlled efficiently.

onAttach()

onCreate()

onCreateDialog()

onCreateView()

onViewCreated()

onDestroy()

**8.       With the help of example explain spinner.**

Android Spinner is a view similar to the dropdown list which is used to select one option from the list of options. It provides an easy way to select one item from the list of items and it shows a dropdown list of all values when we click on it. The default value of the android spinner will be the currently selected value and by using Adapter we can easily bind the items to the spinner objects. Generally, we populate our Spinner control with a list of items by using an ArrayAdapter in our Kotlin file.

Different Attributes for Spinner Widget

android:id    Used to specify the id of the view.

android:textAlignment     Used to the text alignment in the dropdown list.

android:background         Used to set the background of the view.

android:padding     Used to set the padding of the view.

android:visibility    Used to set the visibility of the view.

android:gravity      Used to specify the gravity of the view like center, top, bottom, etc

**9.       Explain GridView Using adapter with the help of example.**

A GridView is a type of AdapterView that displays items in a two-dimensional scrolling grid. Items are inserted into this grid layout from a database or from an array. The adapter is used for displaying this data, setAdapter() method is used to join the adapter with GridView. The main function of the adapter in GridView is to fetch data from a database or array and insert each piece of data in an appropriate item that will be displayed in GridView. This is how the GridView structure looks like. Note that we are going to implement this project using the Java language.

XML Attributes of GridView

android:numColumns: This attribute of GridView will be used to decide the number of columns that are to be displayed in Grid.

android:horizontalSpacing: This attribute is used to define the spacing between two columns of GridView.

android:verticalSpacing: This attribute is used to specify the spacing between two rows of GridView.

**10.     Explain ListView Using adapter with the help of example.**

An adapter actually bridges between UI components and the data source that fill data into UI Component. Adapter holds the data and send the data to adapter view, the view can takes the data from adapter view and shows the data on different views like as spinner, list view, grid view etc.

The ListView and GridView are subclasses of AdapterView and they can be populated by binding them to an Adapter, which retrieves data from an external source and creates a View that represents each data entry.

Android provides several subclasses of Adapter that are useful for retrieving different kinds of data and building views for an AdapterView ( i.e. ListView or GridView). The common adapters are ArrayAdapter,Base Adapter, CursorAdapter,SimpleCursorAdapter,SpinnerAdapter and WrapperListAdapter. We will see separate examples for both the adapters.

ListView Attributes

Sr.No Attribute & Description

1        android:id

This is the ID which uniquely identifies the layout.

2        android:divider

This is drawable or color to draw between list items.

3        android:dividerHeight

This specifies height of the divider. This could be in px, dp, sp, in, or mm.

4        android:entries

Specifies the reference to an array resource that will populate the ListView.

**11.     Explain Photo Gallery Using adapter with the help of example.**

In Android, Gallery is a view that can show items in a center locked, horizontal scrolling list, and hence the user can able to select a view, and then the user selected view will be shown in the center of the Horizontal list. “N” number of items can be added by using the Adapter. The adapter is a bridging component between UI component and data source(It can be an array of items defined in java code or from a database). The items given in the adapter will be shown in the gallery in the example.

<Gallery

  android:id="@+id/languagesGallery"

  android:layout\_width="match\_parent"

  android:layout\_height="wrap\_content"

  android:layout\_marginTop="100dp"

  android:unselectedAlpha="50"

  android:spacing="5dp"

  android:animationDuration="2000"

  android:padding="10dp" />

**12.     Explain Work Thread with example.**

When an application is launched in Android, it creates the primary thread of execution, referred to as the “main” thread. Most thread is liable for dispatching events to the acceptable interface widgets also as communicating with components from the Android UI toolkit. To keep your application responsive, it’s essential to avoid using the most thread to perform any operation which will find yourself keeping it blocked.

Android provides some ways of making and managing threads, and lots of third-party libraries exist that make thread management tons more pleasant. However, with numerous approaches at hand, choosing the proper one are often quite confusing. In this article, you’ll study some common scenarios in Android development where threading becomes essential and a few simple solutions which will be applied to those scenarios and more.

Threading in Android

In Android, you’ll categorize all threading components into two basic categories:

Threads that are attached to an activity/fragment: These threads are tied to the lifecycle of the activity/fragment and are terminated as soon because the activity/fragment is destroyed.

Threads that aren’t attached to any activity/fragment: These threads can still run beyond the lifetime of the activity/fragment (if any) from which they were spawned.

**13.     Explain AsyncTask in detail.**

Android AsyncTask going to do background operation on background thread and update on main thread. In android we cant directly touch background thread to main thread in android development. asynctask help us to make communication between background thread to main thread.

Methods of AsyncTask

onPreExecute() − Before doing background operation we should show something on screen like progressbar or any animation to user. we can directly comminicate background operation using on doInBackground() but for the best practice, we should call all asyncTask methods .

doInBackground(Params) − In this method we have to do background operation on background thread. Operations in this method should not touch on any mainthread activities or fragments.

onProgressUpdate(Progress…) − While doing background operation, if you want to update some information on UI, we can use this method.

onPostExecute(Result) − In this method we can update ui of background operation result.

Generic Types in Async Task

TypeOfVarArgParams − It contains information about what type of params used for execution.

ProgressValue − It contains information about progress units. While doing background operation we can update information on ui using onProgressUpdate().

ResultValue −It contains information about result type.

**14.     Explain Broadcast Receiver with example.**

Broadcast in android is the system-wide events that can occur when the device starts, when a message is received on the device or when incoming calls are received, or when a device goes to airplane mode, etc. Broadcast Receivers are used to respond to these system-wide events. Broadcast Receivers allow us to register for the system and application events, and when that event happens, then the register receivers get notified. There are mainly two types of Broadcast Receivers:

Static Broadcast Receivers: These types of Receivers are declared in the manifest file and works even if the app is closed.

Dynamic Broadcast Receivers: These types of receivers work only if the app is active or minimized.

Creating the Broadcast Receiver:

class AirplaneModeChangeReceiver:BroadcastReceiver() {

       override fun onReceive(context: Context?, intent: Intent?) {

            // logic of the code needs to be written here

      }

}

**15.     Explain life cycle of Services.**

The service life cycle consists of the same four stages at the product life cycle: introduction, growth, maturity and decline. The characteristics of each stage are the same. The only difference lies in the strategies that can be used.

**I INTRODUCTORY STAGE**

A new service or a new form of a current service is said to be in the introductory stage when it is first offered. As with goods, many new services never obtain acceptance by customers and never get past the first stage of the service life cycle. An advantage that services have over goods is that many new services can be introduced on a small scale and expanded if acceptance grows. This small scale introduction reduces the financial risk associated with the introduction, making failure less costly.

**II GROWTH STAGE**

During the growth stage, the industry is growing rapidly. Most firms offering the new service are seeing a positive cash flow.

For eg: a patient can learn about the incubation period for chicken pox by either talking to a nurse or dialing into a vast library of prerecorded tapes.

**III MATURITY STAGE**

During the maturity stage, industry sales level off. Competition becomes very intense since the only way a firm can gain the market share or increase sales is to take them away from a competitor At this stage in the service life cycle, consumers see very few distinguishable characteristics among the various firms in a service industry.

**IV DECLINE STAGE**

During the decline stage, industry sales decline. This sales drop is often due to a new technology that has been developed. For eg, typewriter repair services declined bcoz typewriters were largely replaced by computers which resulted in a need for a computer service technicians and computer programmers.

**16.     Explain Notification with example.**

Notification is a kind of message, alert, or status of an application (probably running in the background) that is visible or available in the Android’s UI elements. This application could be running in the background but not in use by the user. The purpose of a notification is to notify the user about a process that was initiated in the application either by the user or the system. This article could help someone who’s trying hard to create a notification for developmental purposes.

Notifications could be of various formats and designs depending upon the developer. In General, one must have witnessed these four types of notifications:

1. Status Bar Notification (appears in the same layout as the current time, battery percentage)

2. Notification drawer Notification (appears in the drop-down menu)

3. Heads-Up Notification (appears on the overlay screen, ex: Whatsapp notification, OTP messages)

4. Lock-Screen Notification (I guess you know it)

**17.     How to create database in SQLite? Explain with example.**

In SQLite, sqlite3 command is used to create a new SQLite database. You do not need to have any special privilege to create a database.

Syntax Following is the basic syntax of sqlite3 command to create a database: −

$sqlite3 DatabaseName.db

Always, database name should be unique within the RDBMS.

Example If you want to create a new database <testDB.db>, then SQLITE3 statement would be as follows −

$sqlite3 testDB.db

SQLite version 3.7.15.2 2013-01-09 11:53:05

Enter ".help" for instructions

Enter SQL statements terminated with a ";"

sqlite>

The above command will create a file testDB.db in the current directory. This file will be used as database by SQLite engine. If you have noticed while creating database, sqlite3 command will provide a sqlite> prompt after creating a database file successfully.

Once a database is created, you can verify it in the list of databases using the following SQLite .databases command.

sqlite>.databases

seq name file

--- --------------- ----------------------

0 main /home/sqlite/testDB.db

You will use SQLite .quit command to come out of the sqlite prompt as follows −

sqlite>.quit

$

**18.     What is cursor in SQLite?**

Cursor is a Temporary Memory or Temporary Work Station. It is Allocated by Database Server at the Time of Performing DML(Data Manipulation Language) operations on Table by User. Cursors are used to store Database Tables. There are 2 types of Cursors: Implicit Cursors, and Explicit Cursors. These are explained as following below.

Implicit Cursors:

Implicit Cursors are also known as Default Cursors of SQL SERVER. These Cursors are allocated by SQL SERVER when the user performs DML operations.

Explicit Cursors :

Explicit Cursors are Created by Users whenever the user requires them. Explicit Cursors are used for Fetching data from Table in Row-By-Row Manner.

**19.     With the help of example describe getting location data.**

Also known as geographic information or geospatial data, location data refers to information related to objects or elements present in a geographic space or horizon. There are two basic types of location data: vector data and raster data.

Vector: This form uses points, lines, and polygons to represent features such as cities, roads, mountains, and bodies of water that are mapped and stored in geographic information systems (GIS).

Raster: This form uses cells to represent spatial features. An example would be remote satellite data.

Examples of data visualizations that use location data

Technology advances are improving our ability to capture more location data every day. While previously only plotted on a physical map, the list of map types grew as this type of data and different technology (i.e. mobile devices, location-based services, etc.) evolved

**20.     Explain the term displaying Google map in detail.**

Android provides facility to integrate Google map in our application. Google map displays your current location, navigate location direction, search location etc. We can also customize Google map according to our requirement.

Types of Google Maps

There are four different types of Google maps, as well as an optional to no map at all. Each of them gives different view on map. These maps are as follow:

Normal: This type of map displays typical road map, natural features like river and some features build by humans.

Hybrid: This type of map displays satellite photograph data with typical road maps. It also displays road and feature labels.

Satellite: Satellite type displays satellite photograph data, but doesn't display road and feature labels.

Terrain: This type displays photographic data. This includes colors, contour lines and labels and perspective shading.

None: This type displays an empty grid with no tiles loaded.

Syntax of different types of map

googleMap.setMapType(GoogleMap.MAP\_TYPE\_NORMAL);

googleMap.setMapType(GoogleMap.MAP\_TYPE\_HYBRID);

googleMap.setMapType(GoogleMap.MAP\_TYPE\_SATELLITE);

googleMap.setMapType(GoogleMap.MAP\_TYPE\_TERRAIN);