Compulsory question:

***a. Why is Android called a Software stack?***

A software stack is a group of programs that work in tandem to produce a result or achieve a common goal. Software stack also refers to any set of applications that works in a specific and defined order toward a common goal, or any group of utilities or routine applications that work as a set.

b. ***Give any two methods of testing android mobile applications***

Use Android Virtual Device (AVD) known as Emulator.

Use Physical Device (Android Smartphone).

c. ***How can you differentiate a relative layout from a linear layout?***

***Relative layout*** i:t is a view group that displays child view in relative layout.

***Linear layout***: it is a view group that aligns all children in single direction, vertically or horizontally.

d. **As an Android developer, how would you generate money from a mobile application you developed? Give at least 3 ways.**

*Advertising: interstitial, video, native, incentivized, display ads and banners*

*Email marketing*

*Sponsorship*

*Subscriptions*

*In-app purchases*

*Freemium upsell*

*Amazon Underground*

*Physical purchases and merchandise*

*Collecting and selling data*

*Affiliate income and referral marketing (from CPA to CPI)*

e. **Briefly give the role of the following components in android**

I. ***Android 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.*

II. ***Event Listeners:*** *an event listener is an interface in View Class that contains a single callback method.*

*III.* ***onClickListener:*** *is used to detect click style events whereby the user touches. IV.* ***Gradle:*** *compiles app resources and source code into APK file.*

*V.* ***Adapter:*** provides access to the data items.

VI. Dalvik*: is a discontinued process virtual machine (VM) in****Android****operating system that executes applications written for****Android***

***Vii***

**f. Differentiate the following**

I.***Toast and Dialog*** :

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

***Alerts*** in modal dialogs should be used whenever an explicit response is needed from the user, and that response is needed before anything else can happen.

***A dialog*** does not fill the screen and is normally used for modal events that require users to take an action before they can proceed.

**II.Application Layer and API Framework layer**:

***Application:*** composed with system applications and third party applications Android system has some applications that comes with it, these includes: Email, SMS, browser, contact, etc…

**API Framework layer:** it is a set of API built in Java.

• Developers have access to the same API framework used by Android system.

***g. What are the steps of publishing an android app Playstore.***

1. Create an account

2. Familiarize yourself with Developer Console

3. Fill in the necessary account details

4. Link your merchant account

5. Upload your app

6. Alpha and beta testing of app

7. Provide details for store listing

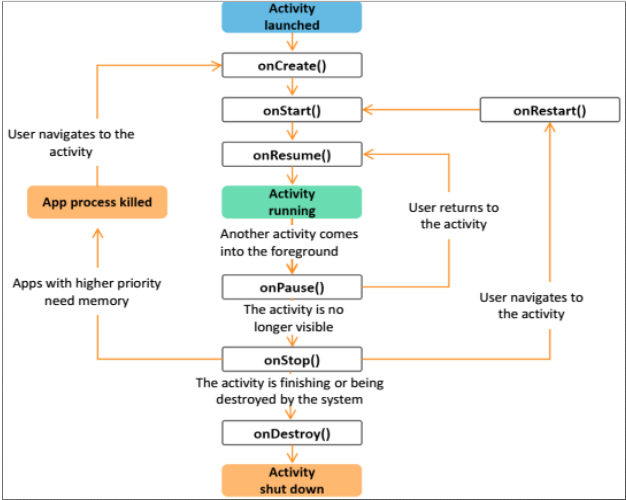
8. Add pricing and distribution details

9. Publishing the application

10. Device Filtering option

**Answer any two questions: !!**

***a. Use a diagram to illustrate the phase of activity lifecycle in mobile Applications***



b. ***Briefly explain an Intent in Android applications***

Intent is a message that activate either a specific component (explicit intent) or specific type of component (implicit).

Intent is abstract description of an operation to be performed.

Intent is most significant used in launching of activities, where it can be thought of as glue between activities.

**Example of two ways Intent can be used**

**Intent can be used in two ways:**

* **From one activity to another in same application**
* **From activity in one app to another app**

**!!**c. Give an example of how an Intent can be used to pass a “Hello” message from the MainActivity.java and start a second activity named Display.java where the ‘hello message will be displayed’.

package com.example.helloworld;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

public class MainActivity extends AppCompatActivity {

EditText editText

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main); findViewById(R.id.button).setOnClickListener(new handleButton()); }

class handleButton implements View.OnClickListener {

public void onClick(View v) {

String name = EditText.getText().toString();

If(name.length() ==0){

editText.setError(“you must enter Hello message”)

}

Else{

Intent intent = new Intent(MainActivity.this,Display.class);

Intent.putExtra(“hello” ,name);//*send and display text to display from MainActivity*

textView.setText(“hello”,name)// *display text in MainActivity from Display*

startActivity(intent);

}

Intent intent = new Intent(MainActivity.this, Display.class);

startActivity(intent);

}

} }

Use example to explain different types of services in Android Applications

**Android has 3 types of services**

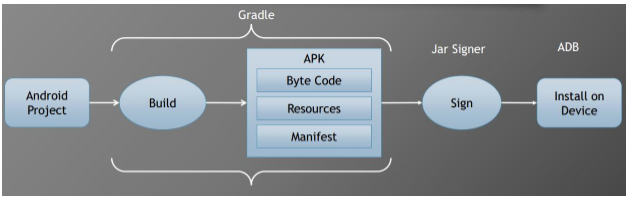
*•* ***Scheduled services:*** *may be initiated by the JobScheduler API. Example: Repeat Alarms*

*•* ***Started Services****: started by an application component like an activity then it runs in the background even when the components that started it is closed. • Example: Downloading a file*

*•* ***Bound services:*** *It allows components (such as activities) to bind to the service, send requests, receive responses, and perform inter-process communication (IPC). It is used when an activity wants to interact directly with the service • Example:*

***Android service:*** *is an application component that can perform long running operations in the background. Ex: play music, perform final IO*

a. With the help of diagram explain the summarized process of developing Android applications.!!



b. Give the purple of the following functions/methods in Android development

***I. onLongClickListener ():*** Used to detect when the user maintains the touch over a view for an extended period.

***II. OnCreate ():*** OnCreate (Bundle savedInstanceState) Function in Android: When an Activity first call or launched then onCreate (Bundle savedInstanceState) method is responsible to create the activity. ... But the reason it is used in app is because that method is the best place to put initialization code.

***III. SetContentView:*** setContentView() is a very important function when it comes to programming with Android. One has to understand its use completely to work with Android User Interface. Basically what this function does is display the Layout created through XML or the Dynamically created layout view in the Screen.

***IV. Async Task ():*** Async Task enables proper and easy use of the UI thread. This class allows you to perform background operations and publish results on the UI thread without having to manipulate threads and/or handlers.

***V.* Realm.init*():*** *replace both SQLite and ORM libraries in your****Android****projects.*

**VI. getWritableDatabase():***Create and/or open a database that will be used for reading and writing.*

***OnItemClickListener():***Callback method to be invoked when an item in AdapterView has been clicked

***OnResume():***  *When the activity enters the Resumed state, it comes to the foreground, and then the system invokes the***onResume()***callback.*

***AlertDialog.Builder():****can be used to display the dialog message with OK and Cancel buttons. It can be used to interrupt and ask the user about his/her choice to continue or discontinue.*

**SetContentView():** SetContentView is used to fill the window with the UI provided from layout file incase of setContentView(R. layout. somae\_file).

**OnProgressUpdate():**invoked on the UI thread after a call to publishProgress(Progress...).

c. write XML code to design a login layout using relative layout as the root element. The layout is going to have username (as email) field, the password and a login button.

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

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

*amdroid:orientation="vertical"*

*android:padding="10dp"*

*android:layout\_width="match\_parent"*

*android:layout\_height="match\_parent">*

*<TextView*

*android:layout\_width="wrap\_parent"*

*android:Text="Username"*

*android:layout\_height="wrap\_content" />*

*<EditText*

*android:id="@+id/etUsername"*

*android:layout\_width="match\_parent"*

*android:layout\_marginBotton="10dp"*

*android:layout\_height="wrap\_content"*

*android:inpuType="textEmail" />*

*<TextView*

*android:layout\_width="wrap\_parent"*

*android:Text="Password"*

*android:layout\_height="wrap\_content"* />

*<EditText*

*android:id="@+id/etPassword"*

*android:layout\_width="match\_parent"*

*android:layout\_marginBotton="10dp"*

*android:layout\_height="wrap\_content"*

*android:inpuType="textPassword" />*

*<Button*

*android:id="@+id/bLogin"*

*android:Text="Login"*

*android:layout\_width="match\_parent"*

*android:layout\_height="wrap\_content" />*

*</RelativeLayout>*

***Good luck***

1**)what is material design in android and what role does it play**

***Material design****is a comprehensive guide for visual, motion, and interaction****design****across platforms and devices.*

*To use****material design****in your****Android****apps, follow the guidelines defined in the****material design****specification and use the new components and styles available in the****material design****support library.*

2)**briefly explain five features of firebase that would be useful in android development**

Here's the list of the features:

***Realtime Database***: is a cloud hosted **database** that supports multiple platforms **Android**, iOS and Web.

***Hosting:*** it provides fast and secure **hosting** for your web app, static and dynamic content, and microservices.

***Authentication***: Is a process of recognizing user identity

***Storage***:  is where you keep data, like music and photos.

***Cloud Messaging:*** is a cross-platform **messaging** solution which reliably sends the message at no cost.

***Remote Config:***  is a cloud service that lets you change the behavior and appearance of your app without requiring users to download an app update.

***Test Lab:*** uses real, production devices running in a Google data center to **test** your app

***Notifications:*** is a message that **Android** displays outside your app's UI to provide the user with reminders, communication from other people, or other timely information from your app.

***Dynamic Links:*** are smart **URLs** that allow you to send existing and potential users to any location within your iOS or Android app.

***Invites:*** it helps to **Invite** Users to Your App.

***AdWords:***  is one of the services advertisers use for online promotion of their content, brand, website,

***AdMob:*** is a mobile advertising platform that you can use to generate revenue from your app.

**3.Briefly differentiate the following concept in android**

***Implicit Intent vs Explicit Intent:***

***Implicit intent:*** is intent that will help you to move from one activity to another but in case you don’t know the destination activity

WHILE

***Explicit Intent*** is an intent that will help you to move from one activity to another but in case you know the destination activity.

***Activity and Fragment***

***Activity****is an application component that gives a user interface where the user can interact. The****fragment****is a part of an****activity****, which contributes its own UI to that****activity****.*

***OnCreate() and OnStart() for android activity***

*is called when the when the****activity****is first created.****onStart()****is called when the****activity****is becoming visible to the user.*

***START\_STRICKY AND START\_NOT\_STRICKY***

***START\_STICKY****tells the OS to recreate the service after it has enough memory and call****onStartCommand****() again with a null intent.****START\_NOT\_STICKY****tells the OS to not bother recreating the service again.*

***4.Briefly explain the essential steps in integrating fire base in android project***

*Add****Firebase****using the****Firebase****console.*

***Step****1: Create a****Firebase project****.*

***Step****2: Register your****app****with****Firebase****.*

***Step****3: Add a****Firebase****configuration file.*

***Step****4: Add****Firebase****SDKs to your****app****.*

*5.Explain different types of broadcast receivers in android APPLICATION*

*A****broadcast****receiver (receiver) is an****Android****component which allows you to register for system or****application****events.*

***There are two types of broadcast receivers:***

*Static****receivers****, which you register in****the Android****manifest file.*

*Dynamic****receivers****, which you register using a context*

*6.Create a model class that would be using to store information about dogs in realm database from android activity .Assume dogs are identify by two characteristics Name and age.*

*package com.example.Realmapp;*

*public class Dogs {*

*public String name, age;*

*public Dogs(){*

*}*

*public Dogs(String Name, String Age ){*

*this.name = Name;*

*this.age = Age;*

*}*

*}*

7.Explain core building blocks of any android app

**Content Providers**

A content provider is a component for managing a data set.

**Broadcast Receivers!!**

**activities** !!

services !!

*8.What do you understand by permission in android applications? And where can a developer set same of those permissions*

***app permissions govern what your app is allowed to do and access. This ranges from access to data stored on your phone, like contacts and media files, through to pieces of hardware like your handset's camera or microphone.***

 In **Android permissions** are declared in AndroidManifest. ...

### **activity\_main.xml**

First we need to drag and drop ListView component from palette to activity\_main.xml file.

**File: activity\_main.xml**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
2. **<Relative layout**
3. android:layout\_width="match\_parent"
4. android:layout\_height="match\_parent"
5. tools:context=".MainActivity"**>**
7. **<ListView**
8. android:id="@+id/listView"
9. android:layout\_width="match\_parent"
10. android:layout\_height="fill\_parent"
11. **/>**
12. **</Relative Layout>**

Create an additional mylist.xml file in layout folder which contains view components displayed in listview.

### **mylist.xml**

**File: mylist.xml**

1. **<?xml** version="1.0" encoding="utf-8"**?>**
3. **<TextView**
4. android:id="@+id/textView"
5. android:layout\_width="wrap\_content"
6. android:layout\_height="wrap\_content"
7. android:text="Medium Text"
8. android:textStyle="bold"
9. android:textAppearance="?android:attr/textAppearanceMedium"
10. android:layout\_marginLeft="10dp"
11. android:layout\_marginTop="5dp"
12. android:padding="2dp"
13. android:textColor="#4d4d4d"
14. **/>**

Now place the list of data in strings.xml file by creating string-array.

### **strings.xml**

**File:strings.xml**

1. **<resources>**
2. **<string** name="app\_name"**>**ListView**</string>**
3. **<string-array** name="array\_technology"**>**
4. **<item>**Android**</item>**
5. **<item>**Java**</item>**
6. **<item>**Php**</item>**
7. **<item>**Hadoop**</item>**
8. **<item>**Perl**</item>**
9. **</string-array>**
10. **</resources>**

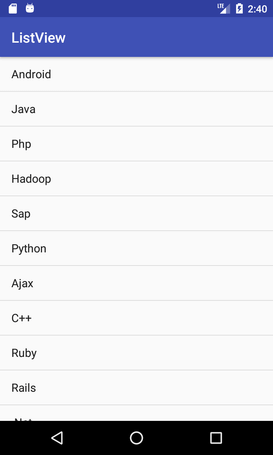
### **Activity class**

In java class we need to add adapter to listview using setAdapter() method of listview.

**File: MainActivity.java**

1. **package** listview.example.com.listview;
2. **import** android.support.v7.app.AppCompatActivity;
3. **import** android.os.Bundle;
4. **import** android.view.View;
5. **import** android.widget.AdapterView;
6. **import** android.widget.ArrayAdapter;
7. **import** android.widget.ListView;
8. **import** android.widget.TextView;
9. **public** **class** MainActivity **extends** AppCompatActivity {
10. ListView listView;
11. TextView textView;
12. String[] listItem;
13. @Override
14. **protected** **void** onCreate(Bundle savedInstanceState) {
15. **super**.onCreate(savedInstanceState);
16. setContentView(R.layout.activity\_main);
18. listView=(ListView)findViewById(R.id.listView);
19. textView=(TextView)findViewById(R.id.textView);
20. listItem = getResources().getStringArray(R.array.array\_technology);
21. **final** ArrayAdapter<String> adapter = **new** ArrayAdapter<String>(**this**,
22. android.R.layout.simple\_list\_item\_1, android.R.id.text1, listItem);
23. listView.setAdapter(adapter);
25. listView.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {
26. @Override
27. **public** **void** onItemClick(AdapterView<?> adapterView, View view, **int** position, **long** l) {
28. // TODO Auto-generated method stub
29. String value=adapter.getItem(position);
30. Toast.makeText(getApplicationContext(),value,Toast.LENGTH\_SHORT).show();
32. }
33. });
34. }
35. }

Output



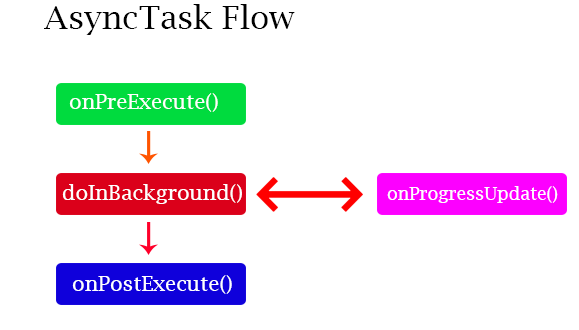
6. **differentiate stopped and Killed state of an activity**

***Stopped****– An****activity****is in a****stopped****state when it's no longer visible.*

***Killed state:*** *the system may***kill***the process at any time to reclaim memory*

*7.* **what is the purpose of a sync task class in android application, and give any two of its overridden methods**

*AsyncTask (Asynchronous****Task****) allows us to run the instruction in the background and then****synchronize****again with our main thread.*

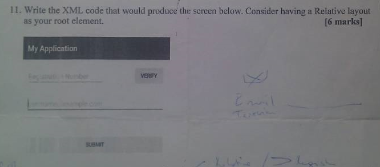


*???8.***Briefy the explain why selecting the minimum SDK is important and what should be considered, when**

*The****min sdk version****is the earliest release of the Android SDK that your application can run on. Usually this is because of a problem with the earlier APIs, lacking functionality, or some other behavioural issue.*

**9. briefly explain how an emulator works**

*The****Android Emulator****simulates****Android****devices on your computer so that you can test your application on a variety of devices and****Android****API levels without needing to have each physical device. The****emulator****provides almost all of the capabilities of a real****Android****device.*



<?xml version="1.0" encoding="utf-8"?>  
<RelativeLayout xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background=""  
 tools:context=".MainActivity">  
  
 <EditText  
 android:id="@+id/etusername"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:hint="enter user name"  
 android:layout\_marginTop="100dp"  
 android:layout\_marginLeft="20dp"  
 android:textSize="24dp"/>  
 <EditText  
 android:id="@+id/password"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:hint="enter user name"  
 android:layout\_below="@+id/etusername"  
 android:layout\_marginTop="100dp"  
 android:layout\_marginLeft="20dp"  
 android:textSize="24dp"/>  
 <Button  
 android:id="@+id/buttonverify"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_toEndOf="@+id/etusername"  
 android:layout\_marginTop="100dp"  
 android:layout\_marginLeft="90dp"  
 android:text="VERIFY"  
 android:textSize="20sp"/>  
 <Button  
 android:id="@+id/buttonSubmit"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_below="@+id/password"  
 android:layout\_marginTop="100dp"  
 android:layout\_marginLeft="90dp"  
 android:text="VERIFY"  
 android:layout\_marginRight="20dp"  
 android:textSize="20sp"  
 />  
  
  
  
  
  
  
</RelativeLayout>