MobiQ&A™

Android Q&A

Exponential Mobile

Exponential Mobile is a developer of mobile content that enables people to exponentiate their potential. The author, Joseph J Kim, can be reached at info@exponentialmobile.com

Table of Contents

8. What are the two types of Intents?
9. What are the actions in intent and what are its two types?
10. How do you start another activity using explicit intent?
11. What are extras in intents?
12. How would you start another Activity requesting a result back?
13. What is setResult() method and how is it used?
14. How would the sending Activity handle the returning result from the receiving Activity?
15. How do you start a Service using intent?
16. How do you start a Broadcast using intent?
17. How do you advertise which implicit intent your app can receive?
18. How would you create an implicit intent for a time alarm?
19. How would you create an implicit intent for sending email?
20. How would you add an implicit intent for adding a calendar event?
21. How would you add implicit intent for taking pictures and store the returning result?
22. How would you create intent when you have a class name only in the form of a string?
23. What are the 6 stages of the Android activity lifecycle?

24. What are the 3 nested loops in an activity lifecycle?
25. Which method of the activity should you save lifecycle changes?
26. Which method of the activity lifecycle kills an activity and purges it from memory?
27. In which method of the activity lifecycle does an activity become invisible?
28. Once an application has been started, which two activity lifecycle is next?
29. Once an application has been resumed, which activity lifecycle is next?
30. Once an application has been paused, which activity lifecycle is next?
31. Once an application has been stopped, which activity lifecycle is next?
32. Once an application has been destroyed, how can it start the activity again?
33. At which part of the activity lifecycle can a user start interfacing with the activity?
34. By default, what is the order of methods executed when one activity starts another activity using intents?

35. How would you go from onResume() to onDestroy() and bypass onPause() and onStop()?
36. What is a backstack of Activities?
37. What is a fragment?
38. Why should anyone create a fragment?
39. How are activity lifecycle of activity and fragments coordinated?
40. What is the fragment lifecycle when it is started and can interact with the user?
41. What is the fragment lifecycle when it needs to be destroyed?
42. Which part of fragment lifecycle is called per each activity's lifecycle?
43. What are the two approaches to hosting a fragment within an Activity?
44. What is a typical way to create a fragment using XML layout?
45. What is the typical way to add a fragment using Java code?
46. What does FragmentManager do?
47. How would you programmatically add, replace, and remove fragments?
48. Why you might want to call addToBackStack() before committing changes to a fragment?
49. What does onCreateView() method in fragment do?

50. What is a bundle?
51. How would you pass bundles from activity to another activity?
52. How would an activity receive a bundle?
53. How would you pass bundles from activity to fragment?
54. How would a fragment receive a bundle?
55. How would you enable Android to run one fragment in portrait and another fragment in landscape modes?
56. How would you create a multi-pane fragment?
57. How would you run public methods in a fragment from an activity?
58. How would you run public methods in an activity from the child fragment?
59. What is a Context in Android?
60. What are some of the most commonly used themes in Android?
61. What is a loader?
62. How do you start a loader?
63. How would you discard old data in a loader?
64. How would you create an onClick listener for a button?
65. How would you create a button in a linear layout using only java code?
66. How would you set up a listener for a button using XML?

67. What is the compact way to create the same listener as above?
68. What is “super” in Android/Java?
69. What is @override in Android/Java
70. How would you capture the back button click?
71. How would you start a web browser with custom URL within an Activity?
72. How would you hide the Android keyboard using a method?
73. How would you remove the action bar?
74. How would you create a class that determines the screen width and height density?
75. How would you retrieve a multiple string values in the resources folder?
76. How would you loop through all the mipmaps in Android and verify if all the mipmaps matches an array?
77. How do you prevent Android from restarting when there is an orientation (portrait to landscape and vice versa)?
78. How would you create a listener for the “Done” button click on a numeric keyboard?
79. How do you create a listener for EditText for text changes?
80. You have just added a Button in your ListView and the the listview listener (ListView.setOnItemClickListener) no longer responds. How would you fix this?
81. How would you obtain the text of a button that is passed on as a View?
82. How would you create a border for a button?

input parameter?

100. How would you create a method to connect to the web and fetch data using HttpURLConnection with a URI as an input parameter using secure username and password?

101. How would you create a class that parses XML?

102. How would you create a class that parses JSON?

103. How would you download an image from the web and display it on an ImageView?

104. What is the difference between get request and post request from web service?

105. In the MVC model, can the controller communicate directly to the Model and View

106. In the MVC Model, can the View communicate directly to the Model?

107. In the MVC model, can the View communicate directly to the controller?

108. What is a synchronized block?

109. How would you implement synchronized block for LinkedList?

110. Why does using synchronized block often cause performance bottlenecks? What is the alternative?

111. What is the difference between ViewGroup and View?

126. How would you create a button that is in shape of a circle?

127. How would you create a button that “floats” (but not moves) on top of a layout using Java code? How would you create its listener?

128. How would you create a toast message?

129. How would you implement autocomplete suggestions in the EditText field?

130. What are some of the widgets available in Android?

131. How would you create checkboxes and handle their clicks?

132. How would you create radio buttons and handle their clicks?

133. How would you implement a toggle button click event?

134. What is a dialog and how would you create it?

135. How would you create a dialog as a fragment?

136. Name the 6 types of event handlers.

137. How would you create menus and handle the menu item click events?

138. Tell me about shared preferences

152. What are different types of logcat methods?

153. What are Adapter Views?

154. What are adapters?

155. How would you create a method that returns the name of a particular country based on location (latitude and longitude)?

156. What are the steps needed to utilize Google Maps?

157. How would you fill your entire activity screen with Google Map?

158. How would you set the latitude and longitude for Google Map with fragment (or xml) layout?

159. How would you set the latitude and longitude for Google Map with java code?

160. How would you get a map of satellite, terrain, hybrid, and normal image with fragment (or xml) layout?

161. How would you get a map of satellite, terrain, hybrid, and normal map image with java code?

162. How do you set markers on Google Map?

163. How would you set markers on Google Map and center it on the screen?

164. How would you create a listener for long click on a Google Map screen?

165. How would you create a listener and method for dragging markers on Google Map?

166. What are the 4 java classes related to sensors?

167. What the 4 different settings for GPS location settings?

168. How would you launch Google GPS Directional Map from an activity?

169. How would you show YouTube videos on your Android app?

170. How would check if there is a cellular network connectivity?

171. What are some of the most commonly used animation methods?

172. How would you animate a View object to slide and fade from left to right?

173. Tell me about 3 different types of animations

174. What are the 3 main subclasses of the Animator class in property animation?

175. What are some of the XML attributes of the Property Animation?

176. What are some of the XML attributes of the View Animation?

177. What is an interpolator?

178. What is overridePendingTransition(R.id.exitAnimation, R.id.enterAnimation)?

179. What is the difference between property animation and view animation?

180. How would you create an Activity that is sensitive to the user touch gestures?

181. What is a class?

182. What is an instantiation?

183. What is the difference between constructors and regular methods?

184. What is polymorphism?

185. What is inheritance?

186. What is encapsulation?

187. What is abstraction?

188. What is a difference between an object and a class?

189. What is method overloading?

190. What is method overriding?

191. What does ‘final” mean when applied to method, class, or variable?

192. When does the return statement not return?

193. Is multiple inheritance allowed in Java?

194. What is an abstract method and class?

195. What is the difference between “==”and equals()?

196. Reverse the elements within array.

197. Randomly shuffle elements within array?

198. Suppose that people enter a room until a pair of people share the birthday. On average, how many people will have to enter the room?

199. Create a method that determines whether an integer is prime.

200. Create a method that takes an integer and outputs its factorial (N!) using recursion.

201. Create a method that takes n number of disks (1,2,3) and outputs all the moves (left, right) for the Tower of Hanoi.

202. Check whether an array of strings is in alphabetical order.

203. Use an insertion sort to sort a string value.

204. Create a class that creates a Stack (LIFO) with push and pop methods.

205. Create a class that creates a Queue (FIFO) with enqueue and dequeue methods.

206. How do you delete a 3rd node in a Queue or Stack?

207. How would you insert a node with item "@" the 3rd element of a Queue or Stack?

208. Create a method that takes a string, and sorts the characters in the string, and returns the sorted characters as a string using Selection Sort.

209. Create a method that takes a string, and sorts the characters in the string, and returns the sorted characters as a string using Insertion Sort.

210. Create a class that takes a string, and sorts the characters in the string, and returns the sorted characters as a string using Merge Sort.

211. Create a class that takes a string, and sorts the characters in the string, and returns the sorted characters as a string using Quick Sort.

212. Create a class for Sequential Search (unordered linked list) with get (fetch value based on key) and put (inserts or update value based on key) that returns first 3 characters.

213. Create a method to determine if a string has all unique characters

214. Create a method to determine if two strings are permutations of the other.

215. Create a method to remove consecutive duplicates from a Stack (LIFO).

216. Create a method that returns the maximum value from a Stack (LIFO).

217. Create a method that returns the minimum value from a Queue (FIFO).

1. What are 4 components of an application?

Activity: Contains most of the code that controls the user interface. An activity has 5 stages in its lifecycle (Create, Start, Resume, Pause, Stop, Destroy). Most android apps focus on the Activity component. An app often contains numerous Activities.

Service: Service runs in the background and does not offer a user interface, and therefore, no user input or output other than through indirect means.

Content Providers: Manages a shared set of map data. SQLite, for example, is a content provider.

BroadCast Receivers: Receives and sends broadcast messages from the system. Broadcast receiver is often just a "gateway" to other components.

2. Why doesn't Android have a single starting point such as main()?

There is no single starting point for Android as there is in many other development platforms in order to allow each component to start from other apps. Android allows 3 (Activities, Services, Broadcast Servers) of the 4 components to start from other applications. This flexible architecture allows one application to run an activity in another application using intents. For example, any Android app can run the camera activity to take pictures which is part of the camera app.

3. What are some of the elements in the manifest (AndroidManifest.xml) file?

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

<manifest>

manifest is the root element of the AndroidManifest.xml file. It has package attribute that describes the package name of the activity class.

<application>

android:icon represents the icon for all the android application components.
android:label works as the default label for all the application components.
android:theme represents a common theme for all the android activities.

<activity>

android:label represents a label i.e. displayed on the screen.

android:name represents a name for the activity class. It is required attribute.

<intent-filter>

intent-filter is the sub-element of activity that describes the type of intent to which activity, service or broadcast receiver can respond to.

<action>

It adds an action for the intent-filter. The intent-filter must have at least one action element.

<category>

It adds a category name to an intent-filter.

4. How do you specify which activity to start (launching activity) when the application runs?

In the manifest file, find the launching activity and change the <category> attribute to end with ".LAUNCHER"

```
<activity
        android:name=".IntentsActivity"
    android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category
android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

5. How would you create a splash screen that lasts for 3 seconds and starts the main activity?

Create the splash screen (splashscreen.xml) layout.

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent">
<ImageView
    android:id="@+id/image"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerInParent="true"
    android:src="@mipmap/ic_launcher" />
</RelativeLayout>
```

Create the java class for the SplashScreen.java activity which will start the target activity (MainActivity) via intent after 3 seconds.

```
public class SplashScreen extends Activity{
private static int SPLASH_SCREEN_DELAY = 3000;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.splashscreen);
new Handler().postDelayed(new Runnable() {
      @Override
      public void run() {
         Intent intent = new Intent(SplashScreen.this, MainActivity.class);
         startActivity(intent);
         finish();     } }, SPLASH_SCREEN_DELAY);    }
  }
```

Configure the AndroidManifest.xml and set SplashScreen.java as the launcher

```
<activity
   android:name=".SplashScreen"
   android:label="@string/app_name" >
<intent-filter>
      <action android:name="android.intent.action.MAIN" />
      <category android:name="android.intent.category.LAUNCHER" />
 </intent-filter>
</activity>
```

6. How do you allow a thread to pause for a few seconds before executing the next line of code?

```
try {
Thread.sleep(2000);
}
catch(InterruptedException e) {
e.printStackTrace();
}
```

7. What is intent?

    a. Intent allows one android app to start a component (usually an Activity) within or outside the app. Activities, Services, Broadcast Servers can be started by intents. Content Provider is not activated by intents but by ContentResolver class.

8. What are the two types of Intents?

The two types of intents are explicit and implicit intents. Explicit intents specify which component an activity should start. Implicit intent does not specify a target component but declare an action to perform which the other components have an option of acting on the call. When you create an implicit intent, the Android system finds the appropriate component to start by comparing the contents of the intent to the *intent filters* declared in the manifest file of other apps on the device. If the intent matches an intent filter, the system starts that component and delivers it the Intent object.

9. What are the actions in intent and what are its two types?

An action in intent specifies an action to perform at the target component. The two types of intent actions are ACTION_VIEW and ACTION_SEND. Both are implemented with startActivity(). ACTION_VIEW is used when you want the target component to view an object, such as a photo. ACTION_SEND is used when you want to send data to the target activity.

10. How do you start another activity using explicit intent?

You can start another activity using by calling the startActivity() method.

```
Intent i=new Intent(CurrentClass.this, TargetClass.class);
startActivity(i);
```

11. What are extras in intents?

Extras in intents are key-value pairs that send information to the target component. putExtra(key,value) method is used by the sending Activity. getExtras() method allows the target Activity to retrieve the key-value pairs sent by the sending Activity. Because the target Activity does not know the datatype of the key-value pair, the target activity must specify the datatype using the following syntax:

Sending Activity

```
Intent i=new Intent(SendingActivity.this,
TargetActivity.class);
i.putExtra("key",50);
startActivity(i);
```

Target Activity

```
Intent i=getIntent()
int targetvalue=i.getExtras().getInt("key");
```

The result should be that “targetvalue” should inherit the integer value of 50.

12. How would you start another Activity requesting a result back?

You can use the startActivityForResult(Intent intent, int requestcode) method if you want the target Activity to return a result. The requestcode is used to identify a intent when there is more than one type of child activity and it is sent back and forth between the sending and receiving Activities. When there is only one receiving Activity, then just 0 would suffice.

```
Intent i=new Intent(CurrentClass.this,
TargetClass.class);
startActivityForResult(i,0);
```

13. What is setResult() method and how is it used?

The setResult() method returns data from the target receiving activity to the sending activity. The method has two types:

```
setResult(int resultCode)
```

This is the simpler version which allows the receiving Activity to send the status of the code execution to the sending Activity. The resultCode is usually RESULT_OK (or -1) or RESULT_CANCELLED (or 0). One can, however, use other

numerals if the result of the execution code is not binary (success or fail).

setResult(int resultCode, Intent i)

This method sends the resultCode as well as data back in the form of intent. The data is specified using putExtra() method (just like the sending Activity). In this example, the receiving Activity is sending back a message back to the sending Activity with the status that the results were OK.

```
Intent intent=new Intent();
intent.putExtra("Sample Message",message);
setResult(RESULT_OK,intent);
  finish();
```

14. How would the sending Activity handle the returning result from the receiving Activity?

You need to create an onActivityResult() method.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent)
{
if (intent==null) return;
else { \\what to do if intent not null }
}
```

The requestCode is the requestCode used by the sending Activity. ResultCode is usually RESULT_OK (or -1) or RESULT_CANCELLED (or 0) and is specified the target receiving activity. “intent” is the intent used by the target receiving Activity to send data back to the sending Activity.

15. How do you start a Service using intent?

    You can start a Service by calling startService() method. You can also bind to the service by using bindService().

16. How do you start a Broadcast using intent?

    You can initiate a Broadcast by calling sendBroadcast(), sendOrderedBroadcast(), or sendStickyBroadcast() methods.

17. How do you advertise which implicit intent your app can receive?

    Receiving implicit intents that are called by other components is configured in the manifest file with the <intent-filter> element. The <intent-filter> specifies how other components may activate the specified component. Inside the <intent-filter> element, there are 3 different types of elements that need to be configured.

    <action>: Declares the intent action accepted, in the name attribute. The value must be the literal string value of an action, not the class constant.

    <data>: Declares the type of data accepted, using one or more attributes that specify various aspects of the data URI (scheme, host, port, path, etc.) and MIME type.

    <category>:Declares the intent category accepted, in the name attribute. The value must be the literal string value of an action, not the class constant.

18. How would you create an implicit intent for a time alarm?

    a. Grant the calling application rights to set alarms by declaring in the manifest file:

       <uses-permission android:name="com.android.alarm.permission.SET_ALARM" />

    b. Use the following method to set the alarm:

```
public void createAlarm(String message, int hour, int minutes) {
Intent intent = new Intent(AlarmClock.ACTION_SET_ALARM)
            .putExtra(AlarmClock.EXTRA_MESSAGE, message)
            .putExtra(AlarmClock.EXTRA_HOUR, hour)
            .putExtra(AlarmClock.EXTRA_MINUTES, minutes);
if (intent.resolveActivity(getPackageManager()) != null) {
      startActivity(intent); }
}
```

       Source: http://developer.android.com/guide/components/intents-common.html

19. How would you create an implicit intent for sending email?

```
   public void composeEmail(String[] addresses, String subject, Uri attachment) {
            Intent intent = new Intent(Intent.ACTION_SEND);
               intent.setType("*/*");
               intent.putExtra(Intent.EXTRA_EMAIL, addresses);
               intent.putExtra(Intent.EXTRA_SUBJECT, subject);
               intent.putExtra(Intent.EXTRA_STREAM, attachment);
               if (intent.resolveActivity(getPackageManager()) != null) {
                  startActivity(intent); }}
```

       Source: http://developer.android.com/guide/components/intents-

common.html

20. How would you add an implicit intent for adding a calendar event?

```
public void addEvent(String title, String location, Calendar begin, Calendar
end) {
                Intent intent = new Intent(Intent.ACTION_INSERT)
                        .setData(Events.CONTENT_URI)
                        .putExtra(Events.TITLE, title)
                        .putExtra(Events.EVENT_LOCATION, location)

.putExtra(CalendarContract.EXTRA_EVENT_BEGIN_TIME, begin)
                        .putExtra(CalendarContract.EXTRA_EVENT_END_TIME,
end);
                if (intent.resolveActivity(getPackageManager()) != null) {
                    startActivity(intent); }}
```

Source: http://developer.android.com/guide/components/intents-common.html

21. How would you add implicit intent for taking pictures and store the returning result?

```
static final int REQUEST_IMAGE_CAPTURE = 1;
    static final Uri mLocationForPhotos;
    public void capturePhoto(String targetFilename) {
        Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        intent.putExtra(MediaStore.EXTRA_OUTPUT,
        Uri.withAppendedPath(mLocationForPhotos, targetFilename);
        if (intent.resolveActivity(getPackageManager()) != null) {
        startActivityForResult(intent, REQUEST_IMAGE_CAPTURE);}}
    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
        if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode ==
RESULT_OK) {
            Bitmap thumbnail = data.getParcelable("data");}}
```

Source: http://developer.android.com/guide/components/intents-common.html

22. How would you create intent when you have a class name only in the form of a string?

```
String targetIntent="com.company.appname.classname";
try{
Class targetClass=Class.forName(targetIntent);
   Intent intent=new Intent(getBaseContext(), targetClass);
   startActivity(intent);
}
catch(Exception e){}
```

23. What are the 6 stages of the Android activity lifecycle?

    Create: started by onCreate()
    Start: started by onStart(); activity becomes visible.
    Resume: started by onResume(); activity becomes visible.
    Pause: started by onPause(); activity becomes partially visible.
    Stop: started by onStop(); activity becomes invisible.
    Destroy: started by onDestroy(); activity is invisible because it no longer is running.

24. What are the 3 nested loops in an activity lifecycle?

    Entire lifetime: describes the lifecycle between onCreate() to onDestroy().

    Visible lifetime: describes the visible lifecycle between onStart() and onStop().

Foreground lifetime: describes the lifecycle between onPause() to onResume()

25. Which method of the activity should you save lifecycle changes?

onPause() is the last activity lifecycle before the Activity is stopped and destroyed. Saving changes during onPause() will guarantee it will take place before the application is stopped.

26. Which method of the activity lifecycle kills an activity and purges it from memory?

onDestroy() method completely kills the activity from memory and it must be created using onCreate() once killed.

27. In which method of the activity lifecycle does an activity become invisible?

onStop() method makes the activity invisible.

28. Once an application has been started, which two activity lifecycle is next?

A started activity can go to either onResume() or onStop() state. An activity cannot go directly from onStart() to onPause().

29. Once an application has been resumed, which activity lifecycle is next?

A resumed activity can go to onPause(). It cannot directly go to create, start, stop, or destroy state.

30. Once an application has been paused, which activity lifecycle is next?

    A paused activity can go to onResume() or onStop(). It cannot create, start, or destroy.

31. Once an application has been stopped, which activity lifecycle is next?

    A stopped activity can go to onRestart() or onDestroy(). It cannot create, resume, or pause.

32. Once an application has been destroyed, how can it start the activity again?

    The destroyed application must go to onCreate(). Once an application is destroyed, all traces of the activity are purged from the memory and cpu.

33. At which part of the activity lifecycle can a user start interfacing with the activity?

    As soon as the activity moves from onStart() to onResume().

34. By default, what is the order of methods executed when one activity starts another activity using intents?

a. Sending Activity A executes onPause().
b. Target receiving Activity B executes onCreate() and then onStart() and then onResume()
c. Sending Activity A executes onStop()

35. How would you go from onResume() to onDestroy() and bypass onPause() and onStop()?

finish() method will cause an activity to go directly from onResume() to onDestroy();

36. What is a backstack of Activities?

Activities are stored as stack (Last In First Out) in Android's memory. Backstack is the name of the activity stack with the activity on the very top, by default, being the only activity in onResume().

37. What is a fragment?

Fragment is a behavior or a dynamic modular user interface within a single activity. There can be multiple fragments within a single activity and each fragment has its own lifecycle, which must be subservient to the parent activity's lifecycle. Fragment allows a single activity to control the layout and behavior using alternative views, properties, methods, models (data), and controls contained in fragments.

38. Why should anyone create a fragment?

Creation of fragments allows an activity to become more dynamic. It allows appearance and disappearance of various

objects. It also allows different user interfaces depending on the user and environmental feedback. Most apps with flexibility will often contain multiple fragments.

39. How are activity lifecycle of activity and fragments coordinated?

Each activity lifecycle callback directly affects similar callbacks in a fragment. For example, an activity's call of onPause() will also result in onPause() call in the fragment that resides in the activity. When an activity is in onResume() state, however, the fragment's lifecycle can be changed independently.

40. What is the fragment lifecycle when it is started and can interact with the user?

onInflate() called when a fragment uses <fragment> tag in the layout.

onAttach(Activity) called once the fragment is associated with its activity. Fragment class has
getActvity() method that can return the attached activity.

onCreate(Bundle) called to do initial creation of the fragment. onCreateView(LayoutInflater, ViewGroup, Bundle) creates and returns the view hierarchy associated with the fragment.

onActivityCreated(Bundle) tells the fragment that its activity has completed its own Activity.onCreate().

onViewStateRestored(Bundle) tells the fragment that all of the saved state of its view hierarchy has been restored.

onStart() makes the fragment visible to the user (based on its containing activity being started).

onResume() makes the fragment interacting with the user (based on its containing activity being resumed). This is when the fragment becomes active.

41. What is the fragment lifecycle when it needs to be destroyed?

    onPause() fragment is no longer interacting with the user either because its activity is being paused or a fragment operation is modifying it in the activity.

    onStop() fragment is no longer visible to the user either because its activity is being stopped or a fragment operation is modifying it in the activity.

    onDestroyView() allows the fragment to clean up resources associated with its View.

    onDestroy() called to do final cleanup of the fragment's state.

    onDetach() called immediately prior to the fragment no longer being associated with its activity.

42. Which part of fragment lifecycle is called per each activity's lifecycle?

    Activity lifecycle of onCreate() calls onAttach(), onCreate(), onCreateView(), and onActivityCreated() methods in the fragment.

    Activity lifecycle of onStart() also calls onStart() method in the fragment.

Activity lifecycle of onResume() also calls onResume() method in the fragment.

Activity lifecycle of onPause() also calls onPause() method in the fragment.

Activity lifecycle of onStop() also calls onStop() method in the fragment.

Activity lifecycle of onDestroy() calls onDestroyView(), onDestroy(), and onDetach() methods in the fragment.

43. What are the two approaches to hosting a fragment within an Activity?

    a. add the fragment to the activity's XML layout
        i. This approach is simple yet inflexible.

    b. add the fragment to the activity's code
        i. By controlling the fragments with code, one can control all aspects of fragments, including its runtime.

44. What is a typical way to create a fragment using XML layout?

    a. Add the fragment to an activity layout

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
```

```
        android:paddingBottom="@dimen/activity_vertical_margin"
        tools:context=".MainActivity">
    <fragment
        android:id="@+id/my_fragment"
      android:name="com.company.appname.MyFragment"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        tools:layout="@layout/my_fragment"
        />
</RelativeLayout>
```

b. Create the layout “my_fragment.xml”

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
            android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" >
    <TextView android:text="@string/hello_world"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content" />
</RelativeLayout>
```

c. Create a java class MyFragment.java extending the layout fragment “my_fragment.xml”

```
public class MyFragment extends Fragment {
   @Override
   public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle
   savedInstanceState) {
        View rootView=inflater.inflate(R.layout.my_fragment, container,
    false );
            return rootView;  }
    }
```

45. What is the typical way to add a fragment using Java code?

a. Create a xml layout called my_fragment.xml

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin" >
<TextView android:text="@string/hello_world"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
</RelativeLayout>
```

b. Create a java class MyFragment.java extending the layout fragment “my_fragment.xml”

```
public class MyFragment extends Fragment {
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
Bundle
savedInstanceState) {
View rootView=inflater.inflate(R.layout.my_fragment, container, false );
return rootView;  }
}
```

c. Create a container (often activity_main.xml)

```
<FrameLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MainActivity"
android:id="@+id/myContainer">
```

```
</FrameLayout>
```

d. Create the fragment in the calling Activity and control it using getFragmentManager().

```
public class CallingActivity extends FragmentActivity {
@Override
public void onCreate(Bundle savedInstanceState){
super.onCreate(savedInstanceState);
MyFragment frag=new MyFragment();
getFragmentManager().beginTransaction().add(R.id.myContainer,
frag).commit();}
    }
```

46. What does FragmentManager do?

FragmentMangager (getFragmentManager()) allows adding, replacing, and removing fragments. FragmentManger is also responsible for adding the fragment views to the activity's stack. To add a fragment to an activity, one should utilize the FragmentManager.

47. How would you programmatically add, replace, and remove fragments?

a. getFragmentManager().beginTransaction().add(R.id.detail Container, fragmentname).commit();

b. getFragmentManager().beginTransaction().replace(R.id.d etailContainer, fragmentname).commit();

c. getFragmentManager().beginTransaction().remove(R.id.d etailContainer, fragmentname).commit();

48. Why you might want to call addToBackStack() before committing changes to a fragment?

If you do not call addToBackStack() when you perform a transaction that removes a fragment, then that fragment is destroyed when the transaction is committed and the user cannot navigate back to it using the back button. Whereas, if you do call addToBackStack() when removing a fragment, then the fragment is *stopped* and will be resumed if the user navigates back using the back button.

```
Fragment newFragment = new ExampleFragment();
FragmentTransaction transaction =
getFragmentManager().beginTransaction();
transaction.replace(R.id.fragment_container, newFragment);
transaction.addToBackStack(null);
transaction.commit();
```

49. What does onCreateView() method in fragment do?

The onCreateView() is called when a fragment is called. It selects a layout (usually XML), inflates the layout, and returns the layout. The onCreateView() has 3 parameters.

LayoutInflator Inflater: Inflates the XML layout file

ViewGroup container: ViewGroup that will contain the fragment.

Bundle SavedInstance: Used to restore the appearance of the fragment.

50. What is a bundle?

Bundle is a key-value pairs generally used for passing data between various activities and fragments. Bundles and

intents are very similar in that they are used to transfer data between activities. Bundles can be sent through multiple intents. The putExtra() key-value pairs in intent, however, are particular to a specific intent.

51. How would you pass bundles from activity to another activity?

    a. Create a method to create a bundle.

```
public Bundle toBundle() {
  Bundle b = new Bundle();
    b.putString("Item1", "Number1");
    b.putInt("Item2", 2);
    b.putDouble("Item3", 3.0);
    return b;
}
```

    b. Use setArgument() to pass the bundle to the fragment.

```
Intent intent = new Intent(getApplicationContext(),
TargetActivity.class);
 intent.putExtras(toBundle());
 startActivity(intent);
```

52. How would an activity receive a bundle?

```
Bundle b = getIntent().getExtras();
String Item1 = b.getString("Item1");
int Item2 = b.getInt("Item2");
Double Item3 = b.getDouble("Item3");
```

53. How would you pass bundles from activity to fragment?

    a. Create a method to create a bundle.

```
public Bundle toBundle() {
  Bundle b = new Bundle();
    b.putString("Item1", "Number1");
    b.putInt("Item2", 2);
    b.putDouble("Item3", 3.0);
    return b;
}
```

b. Use setArgument() to pass the bundle to the fragment.

```
MyFragment frag=new MyFragment();
frag.setArguments(toBundle());
```

54. How would a fragment receive a bundle?

```
Bundle b=getArguments();
String Item1=b.getString("Item1");
int Item2=b.getInt("Item2");
Double Item3=b.getDouble("Item3");
```

55. How would you enable Android to run one fragment in portrait and another fragment in landscape modes?

a. Create separate landscape and portrait layouts for fragments.

i. landscape.xml

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingLeft="@dimen/activity_horizontal_margin"
```

```
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:paddingBottom="@dimen/activity_vertical_margin">
   <TextView
      android:text="This is Landscape"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:textSize="30dp"/>
</RelativeLayout>
```

ii. portrait.xml

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:paddingBottom="@dimen/activity_vertical_margin">
   <TextView
      android:text="This is Portrait"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:textSize="30dp"/>
</RelativeLayout>
```

b. Create separate landscape and portrait java codes

i. FragmentLandscape.java

```
   public class FragmentLandscape extends Fragment{
   @Override
  public View onCreateView(LayoutInflater inflater, ViewGroup container,
     Bundle savedInstanceState) {
  View rootView= inflater.inflate(R.layout.landscape_fragment, container,
false);
  return rootView;}
}
```

ii. FragmentPortrait.java

```
public class FragmentPortrait extends Fragment{
        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup
container,
Bundle savedInstanceState) {
View rootView= inflater.inflate(R.layout.portrait_fragment, container,
false);
return rootView;}
}
```

c. Create a layout that contains the landscape and portrait fragments (activity_main.xml)

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent”
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:paddingBottom="@dimen/activity_vertical_margin"
   tools:context=".MyActivity">
   <FrameLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:id="@+id/fragmentContainer"
      >
</RelativeLayout>
```

d. Use FragmentManager to run either landscape or portrait fragments in the main activity.

```
public class MainActivity extends Activity {
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
FragmentManager fragmentManager = getFragmentManager();
FragmentTransaction fragmentTransaction =
fragmentManager.beginTransaction();
Configuration configInfo = getResources().getConfiguration();
if(configInfo.orientation == Configuration.ORIENTATION_LANDSCAPE){
```

```
FragmentLandscape fragmentLandscape = new FragmentLandscape();
fragmentTransaction.replace(R.id.fragmentContainer,fragmentLandscape
);
}
else {
FragmentPortrait fragmentPortrait = new FragmentPortrait();
fragmentTransaction.replace(R.id.fragmentContainer,fragmentPortrait);
}
fragmentTransaction.commit();
}
```

56. How would you create a multi-pane fragment?

    a. Create separate main and side panel layouts

        i. main_panel.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_red_dark"
    android:gravity="top|center"
    android:orientation="vertical" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Main Panel"
        android:textColor="@android:color/white"
        android:textStyle="bold"
        android:textSize="20sp"/>
</LinearLayout>
```

        ii. side_panel.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_blue_dark"
    android:gravity="top|center"
    android:orientation="vertical" >
```

```
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Side Panel"
        android:textColor="@android:color/white"
        android:textStyle="bold"
       android:textSize="20sp"/>
</LinearLayout>
```

b. Create separate main and side panel java fragment codes

i. MainPanelFragment.java

```
public class MainPanelFragment extends Fragment {
    @Override
  public View onCreateView(LayoutInflater inflater, ViewGroup
container,
    Bundle savedInstanceState) {
      View rootView = inflater.inflate(R.layout.main_panel, container,
false);
      return rootView;
      }
    }
```

ii. SidePanelFragment.java

```
public class SidePanelFragment extends Fragment {
    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
    Bundle savedInstanceState) {
    View rootView = inflater.inflate(R.layout.side_panel, container,
false);
        return rootView;
    }
}
```

c. Create a layout that contains the main and side panel fragments (activity_my.xml)

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/RelativeLayout1"
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:baselineAligned="false"
    android:animateLayoutChanges="true"
    tools:context=".MyActivity" >
    <fragment
        android:id="@+id/main_panel"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        class="com.company.appname.MainPanelFragment" />
    <fragment
        android:id="@+id/side_panel"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        class="com.company.appname.SidePanelFragment" />
</RelativeLayout>
```

d. Use the setVisibility() to show or hide the panels within main activity

```
public class MyActivity extends FragmentActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_my);
        int screenOrientation =
getResources().getConfiguration().orientation;
        if (screenOrientation == Configuration.ORIENTATION_PORTRAIT) {
            hideSidePane();}}
    private void hideSidePane() {
        View sidePane = findViewById(R.id.side_panel);
        if (sidePane.getVisibility() == View.VISIBLE) {
            sidePane.setVisibility(View.GONE);
        }
    }
}
```

57. How would you run public methods in a fragment from an activity?

```
MyFragment fragment = new MyFragment();
fragment.MethodName();
```

58. How would you run public methods in a activity from the child fragment?

```
((NameofActivity)getActivity()).MethodName();
```

59. What is a Context in Android?

As the name suggests, context is the current state (or environmental data) of the application/object. Typically you call it to get information regarding another part of your program. You can get the context by invoking getApplicationContext(), getContext(), getBaseContext() or “this” (when in the activity class).

60. What are some of the most commonly used themes in Android?

Theme.Light: Uses an inverse color schema for background and user elements.
Theme.NoTitleBar.Fullscreen: Removes the title bar and status bar.
Theme.Dialog:Makes an activity look like a dialog.
Theme.Holo.Light: Uses an inverse color scheme with Action Bar.

61. What is a loader?

Loaders are available in every activity and fragment and they provides asynchronous loading of data. Loaders monitor the source of their data and deliver new results when the content changes. Loaders automatically reconnect to the last loader's cursor when being recreated after a configuration change.

62. How do you start a loader?

```
getLoaderManager().initLoader(0, null, this);
```

i. A unique ID that identifies the loader. In this example, the ID is 0.
ii. Optional arguments to supply to the loader at construction (null in this example).
iii. A LoaderManager.LoaderCallbacks implementation, which the LoaderManager calls to report loader events. In this example, the local class implements the LoaderManager.LoaderCallbacks interface, so it passes a reference to itself, this.

63. How would you discard old data in a loader?

You should restart the loader

```
getLoaderManager().restartLoader(0, null, this);
```

64. How would you create an onClick listener for a button?

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
final Button button = (Button)findViewById(R.id.button1);
View.OnClickListener mylistener = new View.OnClickListener() {
```

```
        public void onClick(View v) {
        button.setText("test");
    }};
button.setOnClickListener(mylistener);
}
```

65. How would you create a button in a linear layout using only java code?

```
LinearLayout linearlayout=
(LinearLayout)findViewById(R.id.Linearlayout);
 Button button=new Button(this);
 button.setText("Click Me");
 linearlayout.addView(button);
```

66. How would you set up a listener for a button using XML?

    First setup a onClick property for the button in the xml

    android:onClick="<listener name>";

    Then create a method for the listener in the MainActivity

```
public void <listener name>(View v){ }
```

67. What is the compact way to create the same listener as above?

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
final Button button = (Button)findViewById(R.id.button1);
button.setOnClickListener(new View.OnClickListener() {
    @Override
            public void onClick(View v) {
```

```
            button.setText("test");}
             });
    }
```

68. What is “super” in Android/Java?

    The superclass is the class that the inheriting class was extended from, or null if it wasn't extended. The methods in super class are called with “super.methodname()”.

69. What is @override in Android/Java

    @override allows the inheriting class to override the method of the superclass by modifying the inherited method.

70. How would you capture the back button click?

```
@Override
   public void onBackPressed() {
   //run code
finish();
}
```

71. How would you start a web browser with custom url within an Activity?

```
    Intent browserIntent = new Intent(Intent.ACTION_VIEW,
    Uri.parse(“http:\\www.google.com”));
startActivity(browserIntent);
```

72. How would you hide the Android keyboard using a method?

```
private void hideKeyboard() {
    View view = this.getCurrentFocus();
    if (view != null) {
   InputMethodManager inputManager = (InputMethodManager)
   this.getSystemService(Context.INPUT_METHOD_SERVICE);
   inputManager.hideSoftInputFromWindow(view.getWindowToken(),
   InputMethodManager.HIDE_NOT_ALWAYS);
      }
}
```

73. How would you remove the action bar?

Using Java (If the activity is extending “ActionBarActivity” instead of “Activity”).

```
ActionBar actionBar = getSupportActionBar();
actionBar.hide();
```

Using Manifest and styles: verify that the activity is is not extending “ActionBarActivity”. If so, then change it the “ActionBarActivity” to “Activity”.

Create the following in the \values\styles.xml folder.

```
<style name="Theme.NoTitle"
parent="@android:style/Theme.Holo.Light.NoActionBar"></style>
```

Change the AndroidManifest.xml android:theme setting.

```
android:theme="@style/Theme.NoTitle"
```

74. How would you create a class that determines the screen width and height density?

```
public class ScreenSize {
   private Activity activity;
```

```
private float dpWidth;
private float dpHeight;
public ScreenSize(Activity activity) {
    this.activity = activity;
    Display display = activity.getWindowManager().getDefaultDisplay();
    DisplayMetrics outMetrics = new DisplayMetrics();
    display.getMetrics(outMetrics);
    float density = activity.getResources().getDisplayMetrics().density;
    dpHeight = outMetrics.heightPixels / density;
    dpWidth = outMetrics.widthPixels / density;
}
            public float getWidth() {
    return dpWidth;
}
           public float getHeight() {
    return dpHeight;
}}
```

75. How would you retrieve a multiple string values in the resources folder?

```
String output;
for (int i=0; i<Arrayname.length; i++) {
   Integer temp = getResources().getIdentifier(Arrayname[i], "string",
getPackageName());
   output+=getResources().getString(temp)+"\r\n";
   }
```

76. How would you loop through all the mipmaps in Android and verify if all the mipmaps matches an array?

```
      TextView tv=(TextView)findViewById(R.id.textview);
      String[]
Countries=getResources().getStringArray(R.array.countries_array);

            for (int i=0; i<Countries.length; i++) {
             Integer temp =
getResources().getIdentifier(Countries[i].toLowerCase(), "mipmap",
```

```
getPackageName());
                    if (temp==0) {
          output+=Countries[i]+"\r\n";
       } }
              tv.setText(output);
```

77. How do you prevent Android from restarting when there is an orientation (portrait to landscape and vice versa)?

    Add the following to the AndroidManifest.xml

```
<activity
android:name="com.name.SampleActivity"
android:configChanges="keyboardHidden|orientation|screenSize"
android:icon="@drawable/sample_icon"
android:label="@string/sample_title"
android:screenOrientation="portrait" >
</activity>
```

78. How would you create a listener for the “Done” button click on a numeric keyboard?

```
      EditText editText=(EditText)findViewById(R.id.editText);
      editText.setOnEditorActionListener(new
TextView.OnEditorActionListener() {
      @Override
      public boolean onEditorAction(TextView v, int actionId, KeyEvent event)
{
      if (actionId == EditorInfo.IME_ACTION_DONE) {
      }
        return false;
          }
       });
```

79. How do you create a listener for EditText for text changes?

```
                    editText.addTextChangedListener(new TextWatcher() {
      @Override
```

```
        public void beforeTextChanged(CharSequence s, int start, int count, int
after) {

        }

        @Override
        public void onTextChanged(CharSequence s, int start, int before, int
count) {

        }

        @Override
        public void afterTextChanged(Editable s) {

        }
    });
```

80. You have just added a Button in your ListView and the the listview listener (ListView.setOnItemClickListener) no longer responds. How would you fix this?

    In the layout for the <Button> object, insert the following.

    ```
    android:focusable="false"
    ```

    An alternative way would be to set setFocusable() to false within the Button object.

    ```
    Button button1=(Button) findViewById(R.id.button1);
    button1.setFocusable(false);
    ```

81. How would you obtain the text of a button that is passed on as a View?

    ```
    public void onClick(View view){
    Button b = (Button)view;
    String text = b.getText().toString();
    }
    ```

82. How would you create a border for a button?

Create a button.xml in the /res/drawable folder

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <gradient android:startColor="#FFFFFF"
        android:endColor="#FFFFFF"
        android:angle="270" />
    <corners android:radius="3dp" />
    <stroke android:width="5px" android:color="#eec54d" />
</shape>
```

use the button.xml in the android:background.

```
<Button
    android:id="@+id/resetButton"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_alignParentBottom="true"
    android:gravity="left"
    android:background="@drawable/button"
    />
```

83. How would you create a listener for google map marker click events?

```
mMap.setOnMarkerClickListener(new
GoogleMap.OnMarkerClickListener() {
    @Override
    public boolean onMarkerClick(Marker arg0) {

        \\code here

            return true;
    }
});
```

84. How would you create an implicit intent to send SMS text?

```
Intent sendIntent = new Intent(Intent.ACTION_VIEW);
String  URL="text message”
sendIntent.putExtra("sms_body", URL);
sendIntent.setType("vnd.android-dir/mms-sms");
startActivity(sendIntent);
```

85. Create a method that sends SMS text.

```
private void sendSMS(String phoneNumber, String message)
{
    String SENT = "SMS_SENT";
    String DELIVERED = "SMS_DELIVERED";

    PendingIntent sentPI = PendingIntent.getBroadcast(this, 0,
            new Intent(SENT), 0);

    PendingIntent deliveredPI = PendingIntent.getBroadcast(this, 0,
            new Intent(DELIVERED), 0);

    //---when the SMS has been sent---
    registerReceiver(new BroadcastReceiver(){
        @Override
        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode())
            {
             case Activity.RESULT_OK:
             Toast.makeText(getBaseContext(), "SMS sent",
             Toast.LENGTH_SHORT).show();
                 break;
               case SmsManager.RESULT_ERROR_GENERIC_FAILURE:
                Toast.makeText(getBaseContext(), "Generic failure",
                Toast.LENGTH_SHORT).show();
                   break;
               case SmsManager.RESULT_ERROR_NO_SERVICE:
                   Toast.makeText(getBaseContext(), "No service",
                    Toast.LENGTH_SHORT).show();
                   break;
               case SmsManager.RESULT_ERROR_NULL_PDU:
```

```
                    Toast.makeText(getBaseContext(), "Null PDU",
                     Toast.LENGTH_SHORT).show();
                    break;
                case SmsManager.RESULT_ERROR_RADIO_OFF:
                    Toast.makeText(getBaseContext(), "Radio off",
                     Toast.LENGTH_SHORT).show();
                    break;
            }
        }
    }, new IntentFilter(SENT));

    //---when the SMS has been delivered---
    registerReceiver(new BroadcastReceiver() {
        @Override
        public void onReceive(Context arg0, Intent arg1) {
            switch (getResultCode()) {
                case Activity.RESULT_OK:
                    Toast.makeText(getBaseContext(), "SMS delivered",
                            Toast.LENGTH_SHORT).show();
                    break;
                case Activity.RESULT_CANCELED:
                    Toast.makeText(getBaseContext(), "SMS not delivered",
                            Toast.LENGTH_SHORT).show();
                    break;
            }
        }
    }, new IntentFilter(DELIVERED));

    SmsManager sms = SmsManager.getDefault();
    sms.sendTextMessage(phoneNumber, null, message, sentPI,
deliveredPI);
   }//sendSMS()
```

86. How would you obtain current date and time?

```
String date2 = new SimpleDateFormat("MM-dd-yyyy HH:mm:ss").format(new
Date());
```

87. What is the main thread and how does it relate to AsyncTask?

Each app has a single foreground thread called the main thread. Unfortuantely, the single thread architecture makes it hard to multi-task. Furthermore, there are some tasks which may take few seconds, such as network connection, and you want to do that in a separate thread. The AsyncTask class allows background threads to run simultaneously. In order to connect to the web for data using HTTPRequest, you must use AsyncTask.

88. Why use AsyncTask class and not other concurrent programming tools such as thread, runnable, and java.concurrent?

    Only AsyncTask class allows access to the main thread (usually the calling class) and its methods.

89. What are the objects in the AsynTask class?

```
private class MyTask extends AsyncTask<String, String, String> {

    @Override
    protected void onPreExecute() {

    }

    @Override
    protected String doInBackground(String... params) {

    }

    @Override
    protected void onPostExecute(String result) {


    }

    @Override
    protected void onProgressUpdate(String... values) {
```

```
                }
        }//end AsynTask{}
```

90. What are the 3 generic types of the AsyncTask class?

AsyncTask<params,progress,result>

Params is the input parameter to the AsyncTask class. This value is passed directly from the calling class to to doInBackground(). Params can be passed as an array or as a singular object. Even if the params is not passed as array, it is converted as an array value by the doInBackground() and the first value is always params[0], the second value is params[1] and so on.

Progress specifies the datatype used for the input parameter of onProgressUpdate(). The parameter is usually provided by the doInBackground() when it executes publishProgress().

Result is the datatype of output of the AsynTask class. Result comes from the return value from the doInBackground(). Result is also the same datatype as the input parameter for onPostExecute().

91. What is onPreExecute() of the AsyncTask class?

onPreExecute() is the first method that gets executed and it also has access to methods and
properties of the calling class AsyncTask.

92. What is doInBackground() of the AsyncTask class?

doInBackground() performs the background tasks on a separate thread from the calling class.

This is the method that performs the asynchronous threading. It sends the output of the

asynchronous threading to the onPostExecute(). Unlike onPreExecute(), onPostExecute(), and

onProgressUpdate(), the doInBackground() cannot access the methods of the calling class.

93. What is onProgressUpdate() of the AsyncTask class?

onProgressUpdate() is used to update the status of the AsynTask thread to the calling class.

onProgressDate() has access to the methods of the calling class. Another method called

publishProgress() is used to send parameters to onProgressUpdate() which can use the

information to update the calling class.

94. What is onPostExecute() of the AsyncTask class?

onPostExecute() executes after the doInBackground() and receives the output from

doInBackground(). Like onPreExecute() and onProgressUpdate(), the onPostExecute() has access

to the methods of the calling class.

95. How do you execute tasks through AsyncTask class?

```
MyTask task=new MyTask();
task.execute("inputvalue”);
```

96. How do you execute more than one AsynTask class in parallel (not serial)?

```
MyTask task= new MyTask();
task.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR,
“parameter1”, “parameter2”, “parameter3”);
```

The above code will run 3 AsyncTask in parallel with parameter1, parameter2, and parameter3
as input parameters (params).

97. How would you create a class that accepts context and executes tasks asynchronously?

```
public class MyAsyncTask extends AsyncTask<String, Integer, Integer> {
private Activity mActivity;
public MyAsyncTask(Activity activity) {
    mActivity = activity;
}
@Override
protected void onPreExecute() {
 super.onPreExecute();
    // This will run on the main thread
}
@Override
protected Integer doInBackground(String... inputs) {
    // This will run NOT run on the main thread
```

```
            return 0;
        }
        @Override
        protected void onProgressUpdate(Integer... values) {
            // This will run on the main thread
        }
        @Override
        protected void onPostExecute(Integer i) {
            super.onPostExecute(i);
            // This will run on the main thread
        }
    }
```

98. How would you run AsyncTask tasks?

The below class can be run with `new DownloadFilesTask().execute(url1, url2, url3);`

```
    private class DownloadFilesTask extends AsyncTask<URL, Integer,
Long> {
        protected Long doInBackground(URL... urls) {
        int count = urls.length;
            long totalSize = 0;
        for (int i = 0; i < count; i++) {
        totalSize += Downloader.downloadFile(urls[i]);
        publishProgress((int) ((i / (float) count) * 100));
        if (isCancelled()) break;
        }
        return totalSize;
        }

        protected void onProgressUpdate(Integer... progress) {
        setProgressPercent(progress[0]);
        }
        protected void onPostExecute(Long result) {
        showDialog("Downloaded " + result + " bytes");
        }
        }
```

Source: http://developer.android.com/reference/android/os/AsyncTask.html

99. How would you create a method to connect to the web and fetch data using HttpURLConnection with a string URI as an input parameter?

Create the method.

```
public static String getData(String uri) {
BufferedReader reader=null;
try{
URL url=new URL(uri);
HttpURLConnection con=(HttpURLConnection) url.openConnection();
StringBuilder sb=new StringBuilder();
reader= new BufferedReader(new
InputStreamReader(con.getInputStream()));
String line;
while ((line=reader.readLine()) !=null){
sb.append(line+"\n");
}
return sb.toString();
}
catch(Exception e){
e.printStackTrace();
return null;
}
finally {
if(reader !=null){
try{ reader.close();}
catch(IOException e) { e.printStackTrace(); return null;}
}}}
```

100. How would you create a method to connect to the web and fetch data using HttpURLConnection with a URI as an input parameter using secure username and password?

Create the method

```
public static String getData(String uri, String userName, String password)
{
BufferedReader reader = null;
byte[] loginBytes=(userName+":"+password).getBytes();
StringBuilder loginBuilder=new StringBuilder()
.append("Basic
").append(Base64.encodeToString(loginBytes,Base64.DEFAULT));
try {
URL url = new URL(uri);
HttpURLConnection con = (HttpURLConnection) url.openConnection();
con.addRequestProperty("Authorization", loginBuilder.toString());
StringBuilder sb = new StringBuilder();
reader = new BufferedReader(new
InputStreamReader(con.getInputStream()));
String line;
while ((line = reader.readLine()) != null) {
sb.append(line + "\n");
}
return sb.toString();
}
catch (Exception e) {
e.printStackTrace();
return null;
}
finally {
if (reader != null) {
try {
reader.close();
}
catch (IOException e) {
e.printStackTrace();
return null;
}}}}
```

Call the getData() using the following in AsyncTask class with the string uri as the params[0].

```
MyTask mytask=new MyTask();
mytask.execute(<website for in uri>);
private class MyTask extends AsyncTask<String, String, String> {
@Override
protected String doInBackground(String... params) {
String content =getData(params[0], “username”,”password”);
```

```
return content;
}}
```

101. How would you create a class that parses XML?

Create a model class and create an XML document which matches its variables.

```
public class Butterfly {
        private int productId;
        private String name;
        private String category;
        private String instructions;
        private String photo;
        public int getProductId() {
        return productId;
            }
public void setProductId(int productId) {
                this.productId = productId;
        }
        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public String getCategory() {
                return category;
        }
        public void setCategory(String category) {
                this.category = category;
        }
        public String getInstructions() {
                return instructions;
        }
        public void setInstructions(String instructions) {
                this.instructions = instructions;
        }
        public String getPhoto() {
                return photo;
        }
        public void setPhoto(String photo) {
```

```
                this.photo = photo;
        }
        }
```

Create a XML parser class that returns List<Butterfly> with the XML content in String as input

```
public class ButterflyXMLParser {
public static List<Butterfly> parseFeed(String content) {
try {
    boolean inDataItemTag = false;
    String currentTagName = "";
    Butterfly butterfly = null;
    List<Butterfly> butterflyList = new ArrayList<>();
XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
    XmlPullParser parser = factory.newPullParser();
    parser.setInput(new StringReader(content));
    int eventType = parser.getEventType();
    while (eventType != XmlPullParser.END_DOCUMENT) {
    switch (eventType) {
        case XmlPullParser.START_TAG:
        currentTagName = parser.getName();
        if (currentTagName.equals("product")) {
            inDataItemTag = true;
             butterfly = new Butterfly();
             butterflyList.add(butterfly);
        }
        break;
        case XmlPullParser.END_TAG:
        if (parser.getName().equals("product")) {
            inDataItemTag = false;
        }
        currentTagName = "";
        break;
        case XmlPullParser.TEXT:
        if (inDataItemTag && butterfly != null) {
            switch (currentTagName) {
                case "productId":
                    butterfly.setProductId(Integer.parseInt(parser.getText()));
                    break;
                case "name":
                            butterfly.setName(parser.getText());
                            break;
```

```
                        case "instructions":
                            butterfly.setInstructions(parser.getText());
                            break;
                        case "category":
                            butterfly.setCategory(parser.getText());
                            break;
                        case "photo" :
                                    butterfly.setPhoto(parser.getText());
                        default:
                            break;
                    }
                  }
                  break;
             }
            eventType = parser.next();
          }
          return butterflyList;
      }
              catch (Exception e) {
                 e.printStackTrace();
                 return null;
                 }
      }}
```

102. How would you create a class that parses JSON?

Create a model class and create an JSON document which matches its variables

```
    public class Butterfly {
            private int productId;
            private String name;
            private String category;
            private String instructions;
            private String photo;
            public int getProductId() {
            return productId;
                }
    public void setProductId(int productId) {
                        this.productId = productId;
            }
            public String getName() {
```

```
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public String getCategory() {
                return category;
        }
        public void setCategory(String category) {
                this.category = category;
        }
        public String getInstructions() {
                return instructions;
        }
        public void setInstructions(String instructions) {
                this.instructions = instructions;
        }
        public String getPhoto() {
                return photo;
        }
        public void setPhoto(String photo) {
                this.photo = photo;
        }
        }
```

Create a JSON parser class that returns List<Butterfly> with the JSON content in String as input.

```
        public class ButterflyJSONParser {
        public static List<Butterfly> parseFeed(String content) {
        try {
        JSONArray ar = new JSONArray(content);
        List<Butterfly> butterflyList = new ArrayList<>();
    for (int i = 0; i < ar.length(); i++) {
        JSONObject obj = ar.getJSONObject(i);
                                                        Butterfly butterfly =
new Butterfly();
    butterfly.setProductId(obj.getInt("productId"));
    butterfly.setName(obj.getString("name"));
    butterfly.setCategory(obj.getString("category"));
    butterfly.setInstructions(obj.getString("instructions"));
```

```
butterfly.setPhoto(obj.getString("photo"));
butterflyList.add(butterfly);
}
return butterflyList;
    }
    catch (JSONException e) {
e.printStackTrace();
return null;
}}}
```

103. How would you download an image from the web and display it on an ImageView?

Create a ImageView in the layout

```
<ImageView
  android:id="@+id/img"
  android:layout_width="wrap_content"
   android:layout_height="wrap_content" />
```

Grant the app the permission to access the internet in the manifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.exponentialmobile.test2" >
    <uses-permission android:name="android.permission.INTERNET"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                 <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
```

```
    </application>
</manifest>
```

Create a class that extends AsynTask class

```
public class ImageTask extends AsyncTask<String, Void, Bitmap> {
        ImageView bmImage;
        public ImageTask(ImageView bmImage) {
        this.bmImage = bmImage;
         }
         protected Bitmap doInBackground(String... urls) {
            String urldisplay = urls[0];
             Bitmap bitmap = null;
             try {
                 InputStream in = new java.net.URL(urldisplay).openStream();
                 bitmap = BitmapFactory.decodeStream(in);
             }
          catch (Exception e) {
                 Log.e("Error", e.getMessage());
                 e.printStackTrace();
             }
             return bitmap;
         }
         protected void onPostExecute(Bitmap result) {
          bmImage.setImageBitmap(result);
         }
    }
```

Call the ImageTask class

```
ImageView img=(ImageView)findViewById(R.id.img);
ImageTask MyImage=new ImageTask(img);
MyImage.execute("http://www.hyjeong.com/gr_logo_img/Exponential-
Mobile.jpg");
```

104. What is the difference between get request and post request from web service?

Get request: Parameters are appended to the URI. For fetching data

Post request: Parameters are appended to the request. For inserting data.

Put: For updating data.

Delete: For deleting data.

105. In the MVC model, can the controller communicate directly to the Model and View

Yes, the controller can initiate and communicate directly to both Model and View.

106. In the MVC Model, can the View communicate directly to the Model?

No, neither Model nor View can initiate communicate with each other.

107. In the MVC model, can the View communicate directly to the controller?

Yes, but the view must have an object, such as “onClick” to initiate communication with the Controller.

108. What is a synchronized block?

Synchronized blocks allow a single object to have only one thread executing inside them. All other threads attempting to enter the synchronized block are blocked until the thread inside the synchronized block exits the block.

```
public void add(int value){
   synchronized(this){
      this.count += value;
   }
 }
```

109. How would you implement synchronized block for LinkedList?

```
LikedBlockingQueue<String> blockingQueue=new
LinkedBlockingQueue<String>();
```

110. Why does using synchronized block often cause performance bottlenecks? What is the alternative?

Using ReentrantReadWriteLock allows read only access from multiple concurrent threads on a single objects. Although it is not as exclusive as synchronized blocks, using ReentrantReadWriteLock allows more concurrency because it also allows reads by other threads.

111. What is the difference between ViewGroup and View?

ViewGroup defines the layout of the UI and can hold either other ViewGroup or View. A View lays the UI object on the screen such as TextView and Button that is visible to the user. ViewGroup consists of linear, relative, list view, and grid view layouts. ViewGroup is the metadata and View is something the users see.

112. What is the difference between wrap_content and match_parent in the XML layouts?

wrap_content wraps around the content within the View and is only large enough to contain the content. match_parent tells the view to become the largest size that the ViewGroup allows.

113. What is a layout parameter?

layout parameters are the layout attributes that starts with “layout_”. Although set within the child elements, the layout parameters give instruction to the parent layout on how to arrange the child elements.

114. What is the difference between gravity and gravity_layout?

gravity: sets the gravity of the content of the View it’s used on.

layout_gravity: sets the gravity of the parent layout.

115. What is the difference between margins and padding?

Margins are layout parameters and determine the distance between child elements. Padding is also set within the child elements and provides instructions on how the child element should arrange itself and not in relation to other child elements.

116. What is the Linear Layout ViewGroup?

It is a layout that organizes Views and children ViewGroups into a single horizontal (default) or vertical row. The

Linear Layout can have gravity attribute which includes left, center right for horizontal layouts and top, center, or bottom for vertical layouts.

117. What is layout_weight in Linear layout?

The layout_weight assigns how much space a particular view should take in a Linear Layout. The layout_weight is relative, meaning that each weight assigned is compared against each other and the available space is divided accordingly to their values.

118. What is a Frame Layout View Group?

A frame layout typically displays a single child view. If you choose to add more than one child view to the frame layout, then each view will be stacked on top of each other.

119. What is a Relative Layout ViewGroup?

It is a layout that organizes Views and children ViewGroups relative to each other (left, right, top, bottom). It uses properties such as layout_toRightOf, layout_toLeftOf, layout_blow , and layout_above parameters to relatively place the views with each other. You can also use layout_alignRight, layout_alignLeft, layout_alignTop, or layout_alignBottom to align views with one other.

120. What attributes would you use to place views in relation to the parent layout in Relative Layout?

You can use layout_centerHorizontal or layout_centerVertical attributes. To align a view to the top or bottom of its parent layout you can use layout_alignParentTop, layout_alignParentBottom,layout_alignParentLeft, or layout_alignParentRight.

121. What is a List View ViewGroup?

List view uses an adapter, which retrieves and binds rows of data to the List View, to display a single column of list with one or more rows.

122. What is a Grid View ViewGroup?

Grid view uses an adapter, which retrieves and binds rows of data to the Grid View, to display a multiple columns list with one or more rows.

123. How would you list items with ListView using arrays?

a. Create a <ListView> called listview1.xml.

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" >
    <ListView
        android:layout_width="wrap_content"
        android:layout_height="200dp"
        android:id="@+id/listview1">
```

```
</ListView>
</RelativeLayout>
```

b. Create an array of String.

```
String[] myStringArray=new String[];
```

c. Create an adapter using the ArrayAdapter array data type and bind array data to the adapter.

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>
(this, android.R.layout.simple_list_item_1, myStringArray);
```

d. Create the ListView object

```
ListView listView = (ListView) findViewById(R.id.listview1);
```

e. Bind the adapter to the ListView

```
listView.setAdapter(adapter);
```

124. How would you list items with ListView using ArrayList?

a. Create a <ListView> called listview1.xml.

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
   android:layout_height="match_parent"
     android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" >
   <ListView
    android:layout_width="wrap_content"
```

```
    android:layout_height="200dp"
    android:id="@+id/listview1">
</ListView>
</RelativeLayout>
```

b. Create an array of String.

```
ArrayList<String> myArrayList=new ArrayList<String>;
```

c. Create an adapter using the ArrayAdapter array data type and bind array data to the adapter.

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>
(this,
android.R.layout.simple_list_item_1, myArrayList);
```

d. Create the ListView object

```
ListView listView = (ListView) findViewById(R.id.listview1);
```

e. Bind the adapter to the ListView

```
listView.setAdapter(adapter);
```

125. How would you create a listener for a ListView items?

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> parent, View view, int position,
long id) {

  }}
);
```

126. How would you create a button that is in shape of a circle?

Create the xml file under /drawable/round.xml

```
<?xml version="1.0" encoding="utf-8"?> <shape
xmlns:android="http://schemas.android.com/apk/res/android"
android:shape="oval"> <solid android:color="#9F2200"/> <stroke
android:width="2dp" android:color="#fff" /> </shape>
```

Reference the round.xml in the layout

```
<Button android:layout_width="50dp" android:layout_height="50dp"
android:gravity="center_vertical|center_horizontal" android:text="hello"
android:textColor="#fff" />
```

127. How would you create a button that “floats” (but not moves) on top of a layout using Java code? How would you create its listener?

```
    RelativeLayout containerLayout = new RelativeLayout(this);
    containerLayout.setLayoutParams(new

RelativeLayout.LayoutParams(RelativeLayout.LayoutParams.MATCH_PARENT
,
    RelativeLayout.LayoutParams.MATCH_PARENT));
    ImageView buttonImage=new ImageView(this);
    buttonImage.setImageResource(R.mipmap.ic_launcher);
    RelativeLayout.LayoutParams buttonParams = new
RelativeLayout.LayoutParams(
    RelativeLayout.LayoutParams.WRAP_CONTENT ,
    RelativeLayout.LayoutParams.WRAP_CONTENT );
    buttonParams.addRule(RelativeLayout.ALIGN_PARENT_TOP);
    buttonParams.addRule(RelativeLayout.ALIGN_PARENT_RIGHT);
    buttonParams.height=180;
    buttonParams.width=150;
    buttonImage.setLayoutParams(buttonParams);
    containerLayout.addView(buttonImage);
    addContentView(containerLayout, new RelativeLayout.LayoutParams(
    RelativeLayout.LayoutParams.MATCH_PARENT ,
```

```
        RelativeLayout.LayoutParams.MATCH_PARENT ) );
        buttonImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

            }
        });
```

128. How would you create a toast message?

```
Toast.makeText(getApplicationContext(), "Toast
Message",Toast.LENGTH_LONG).show();
```

129. How would you implement autocomplete suggestions in the EditText field?

Create a <AutoCompleteTextView> called country.xml

```
<AutoCompleteTextView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/autocomplete_country"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content" />
```

Create a string based array in the resources in the strings.xml

```
    <resources>
      <string-array name="countries_array">
          <item>USA</item>
          <item>Canada</item>
          <item>Japan</item>
          <item>Australia</item>
          <item>China</item>
          <item>India</item>
          <item>Korea</item>
          <item>England</item>
        </string-array>
    </resources>
```

a. Bind the string array with the adapter show it in the AutoCompleteTextView in the main java code (typically MainActivity.java)

```
AutoCompleteTextView textView = (AutoCompleteTextView)
findViewById(R.id.autocomplete_country);
    String[] countries =
getResources().getStringArray(R.array.countries_array);
    ArrayAdapter<String> adapter =
                new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, countries);
    textView.setAdapter(adapter);
```

130. What are some of the widgets available in Android?

a. Android Button
   i. Let's learn how to perform event handling on button click.
b. Android Toast
   i. Displays information for the short duration of time.
c. ToggleButton
   i. It has two states ON/OFF.
d. CheckBox
   i. Let's see the application of simple food ordering.
e. AlertDialog
   i. AlertDialog displays a alert dialog containing the message with OK and Cancel buttons.
f. Spinner
   i. Spinner displays the multiple options, but only one can be selected at a time.
g. AutoCompleteTextView
   i. Let's see the simple example of AutoCompleteTextView.
h. RatingBar
   i. RatingBar displays the rating bar.

i. DatePicker
    i. Datepicker displays the datepicker dialog that can be used to pick the date.
j. TimePicker
    i. TimePicker displays the timepicker dialog that can be used to pick the time.
k. ProgressBar
    i. ProgressBar displays progress task.

131. How would you create checkboxes and handle their clicks?

a. Create a checkbox objects with the same onClick name

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <CheckBox android:id="@+id/checkbox1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Check Box 1"
        android:onClick="onCheckboxClicked"/>
    <CheckBox android:id="@+id/checkbox2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Check Box 2"
        android:onClick="onCheckboxClicked"/>
</LinearLayout>
```

b. Create a method for the onClick in the java code.

```
public void onCheckboxClicked(View view) {
  switch(view.getId()) {
  case R.id.checkbox1:
  Toast.makeText(getApplicationContext(), "You have clicked Checkbox
1",
   Toast.LENGTH_LONG).show();
```

```
          break;
          case R.id.checkbox2:
        Toast.makeText(getApplicationContext(), "You have clicked Checkbox 2",
        Toast.LENGTH_LONG).show();
           break;}
          }
```

132. How would you create radio buttons and handle their clicks?

Create radio objects with the same onClick name

```
          <RadioGroup
xmlns:android="http://schemas.android.com/apk/res/android"
             android:layout_width="fill_parent"
             android:layout_height="wrap_content"
             android:orientation="vertical">
             <RadioButton android:id="@+id/radio1"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Radio Button 1"
                android:onClick="onRadioButtonClicked"/>
             <RadioButton android:id="@+id/radio2"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:text="Radio Button 2"
                android:onClick="onRadioButtonClicked"/>
          </RadioGroup>
```

Create a method for the onClick in the java code.

```
             public void onRadioButtonClicked(View view) {
              switch(view.getId()) {
               case R.id.radio1:
Toast.makeText(getApplicationContext(), "You have checked Radio
1",Toast.LENGTH_LONG).show();
                break;
                case R.id.radio2:
Toast.makeText(getApplicationContext(), "You have checked Radio
2",Toast.LENGTH_LONG).show();
                break;}}
```

133. How would you implement a toggle button click event?

Create a Toggle Button view

```
<ToggleButton
                android:id="@+id/togglebutton"
        android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:textOn="Vibrate on"
          android:textOff="Vibrate off"
         android:onClick="onToggleClicked"/>
```

Create a method in the Activity that handles the click event

```
public void onToggleClicked(View view) {
   boolean on = ((ToggleButton) view).isChecked();
      if (on) {    }
      else {    }
   }
```

a. Source: http://developer.android.com/guide/topics/ui/controls/togglebutton.html

134. What is a dialog and how would you create it?

A dialog is a small popup window that prompts the user to make a decision or enter additional information. 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. The most efficient way to create a dialog is to use the AlertDialog class.

```
AlertDialog.Builder alertDialog = new AlertDialog.Builder(this);
alertDialog.setMessage("Testing Dialog");
alertDialog.setPositiveButton("OK",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
```

```
                    dialog.cancel();
                }
            });
        alertDialog.show();
```

135. How would you create a dialog as a fragment?

Create a separate dialog MyDialogFragment.java

```
public class MyDialogFragment extends DialogFragment{
@Override
public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder dialog = new AlertDialog.Builder(getActivity());
        dialog.setTitle("Sample Dialog");
        dialog.setMessage("Hello");
        dialog.setPositiveButton("OK", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialogInterface, int i) {
                Toast.makeText(getActivity(), "Clicked OK",
              Toast.LENGTH_SHORT).show();
                        }});
        dialog.setNegativeButton("CANCEL", new
    DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        Toast.makeText(getActivity(), "Clicked Cancel",
Toast.LENGTH_SHORT).show();
    }});
    return dialog.create();}
      }
```

Call the MydialogFragment.java in the main java class.

```
MyDialogFragment myFragment = new MyDialogFragment();
myFragment.show(getFragmentManager(), "Test");
```

136. Name the 6 types of event handlers.

a. onClick()
    i. From View.OnClickListener. This is called when the user either touches the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses the suitable "enter" key or presses down on the trackball.
b. onLongClick()
    i. From View.OnLongClickListener. This is called when the user either touches and holds the item (when in touch mode), or focuses upon the item with the navigation-keys or trackball and presses and holds the suitable "enter" key or presses and holds down on the trackball (for one second).
c. onFocusChange()
    i. From View.OnFocusChangeListener. This is called when the user navigates onto or away from the item, using the navigation-keys or trackball.
d. onKey()
    i. From View.OnKeyListener. This is called when the user is focused on the item and presses or releases a hardware key on the device.
e. onTouch()
    i. From View.OnTouchListener. This is called when the user performs an action qualified as a touch event, including a press, a release, or any movement gesture on the screen (within the bounds of the item).
f. onCreateContextMenu()
    i. From View.OnCreateContextMenuListener. This is called when a Context Menu is being built (as the result of a sustained "long click"). See the discussion on context menus in the Menus developer guide.

Source: http://developer.android.com/guide/topics/ui/ui-events.html

137. How would you create menus and handle the menu item click events?

Create the menu in the /resource/menu/menu_main.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
<item android:id="@+id/action_search"
      android:title="Item1"/>
   <item android:id="@+id/action_compose"
      android:title="Item2" />
</menu>
```

Populate the menu using onCreateOptionMenu()

```
public boolean onCreateOptionsMenu(Menu menu) {
getMenuInflater().inflate(R.menu.menu_main, menu);
return true;  }
```

Use the onOptionsItemSelected() method to handle the menu events

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
    case R.id.action_search:
          newGame();
          return true;
     case R.id.action_compose:
          showHelp();
          return true;
       default:
          return super.onOptionsItemSelected(item); }
}
```

Source: http://developer.android.com/guide/topics/ui/menus.html

138. Tell me about shared preferences

Shared preferences are key/value pairs that is stored on disk that can later be retrieved. The key must be of String data type. The values must be boolean, int, long, float, or String data type. Shared preferences are a simple and effective way to store and retrieve data.

139. How would you create a new shared preference?

getSharedPreferences(): Use this if you need multiple shared preference files identified by name, which you specify with the first parameter. You can call this from any Context in your app.

getPreferences(): Use this from an Activity if you need to use only one shared preference file for the activity. Because this retrieves a default shared preference file that belongs to the activity, you don't need to supply a name.

Source: http://developer.android.com/training/basics/data-storage/shared-preferences.html

140. What are the different context modes of shared preferences?

MODE_PRIVATE: indicates that the data is private to the current application.

MODE_WORLD_READABLE: Makes the data readable by other applications. This has been deprecated in API 17 for security reasons.

MODE_WORLD_WRITEABLE: makes the data readable and writeable by other applications. This mode also has been depreciated since API 17 for security reasons.

141. How do you store values to preferences and save them?

The shared preferences editor must be instantiated

```
private SharedPreferences
settings=getPreferences(MODE_PRIVATE);
private SharedPreferences.Editor editor=settings.edit();
```

The editor will need to call at least one of the following methods to insert key/values.

```
 editor.putBoolean(String key, boolean value);
   editor.putFloat(String key, float value);
   editor.putInt(String key, int value);
   editor.putLong(String key, long value);
   editor.putString(String key, String value);
   editor.putStringSet(String key, Set<String> value);
```

The editor must commit

```
   editor.commit();
```

142. How do you retrieve values from preferences?

Shared preference must be instantiated

```
   private SharedPreferences settings=getPreferences(MODE_PRIVATE);
```

The setting must use one of the following methods to retrieve values. The default value represents the values which the method will retrieve if no value can be found in the shared preferences.

```
setting.getBoolean(String key, boolean defaultvalue);
```

```
setting.getFloat(String key,float defaultvalue);
setting.getInt(String key, int defaultvalue);
setting.getLong(String key, long defaultvalue);
setting.getString(String key, String defaultvalue);
setting.getStringSet(String key, set<String> defaultvalue);
```

143. How would you remove key/values from preferences?

The shared preference editor must be instantiated

```
private SharedPreferences
settings=getPreferences(MODE_PRIVATE);
private SharedPreferences.Editor editor=settings.edit();
```

Remove key and its corresponding value by using remove() method.

```
editor.remove(String key);
```

Commit the editor

```
editor.commit();
```

144. How would you clear all contents of a preference file?

The shared preference editor must be instantiated

```
private SharedPreferences
settings=getPreferences(MODE_PRIVATE);
private SharedPreferences.Editor editor=settings.edit();
```

Clear all key/value pairs using clear()

```
editor.clear();
```

Commit the editor

```
editor.commit();
```

145. How would you iterate through shared preferences?

```
SharedPreferences prefs =  getSharedPreferences("AppName",
MODE_PRIVATE);
        Map<String,?> keys = prefs.getAll();
        for(Map.Entry<String,?> entry : keys.entrySet()){
         Log.e("map values",entry.getKey() + ": " +
          entry.getValue().toString());
        }
```

146. How to iterate through list in java?

```
   List<Int> list=ArrayList<>();
   for (Int element : list) {
   }
```

147. Iterate through hashmap?

```
   Map<Integer, Integer> map = new HashMap<Integer, Integer>();
   for (Map.Entry<Integer, Integer> entry : map.entrySet()) {
System.out.println("Key = " + entry.getKey() + ", Value = " +
entry.getValue());
   }
```

148. How would you sort a List (ArrayList) of class object Contacts?

Create class for Contacts.

```
public class Contacts {

    private String phoneNumber;
    private String contactName;

    public Contacts (String contactName, String phoneNumber){
        this.phoneNumber=phoneNumber;
        this.contactName=contactName;
    }


    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public String getContactName() {
        return contactName;
    }

    public void setContactName(String contactName) {
        this.contactName = contactName;
    }
}
```

Create a List for Contacts called contactsList

```
List<Contacts> contactsList=new ArrayList<>();
```

Populate the contactsList

```
contactsList.add(new Contacts(“value1”,”value2”));
contactsList.add(new Contacts(“value3”,”value4”));
```

Sort the contactsList using Collections.sort()

```
Collections.sort(contactsList, new Comparator<Contacts>() {
        public int compare(Contacts c1, Contacts c2) {
              return c1.getContactName().compareTo(c2.getContactName());
           }
        });
```

149. How would you reverse a List (ArrayList) of class object Contacts?

```
Collections.sort(contactsList, new Comparator<Contacts>() {
public int compare(Contacts c1, Contacts c2) {
return c2.getContactName().compareTo(c1.getContactName());
} });
```

150. How would you run a code that will run only when the user is running the application for the very first time?

Create the following code in onCreate() of the launcher activity.

```
SharedPreferences prefs =
getSharedPreferences("APPNAME",MODE_PRIVATE);
          boolean previouslyStarted = prefs.getBoolean("previouslyStarted",
false);
          if(!previouslyStarted){
              //run the Code
             SharedPreferences.Editor edit = prefs.edit();
    edit.putBoolean("previouslyStarted", true);
              edit.commit();  }
```

151. How would you guarantee the saving of data to shared preference before the application is stopped or destroyed?

The onPause() method is the last method that is guaranteed to executed prior to an application is destroyed. In

order to reverse the changes made to the shared previous in the previous question:

```
@Override
   public void onPause(){
   super.onPause();
SharedPreferences prefs =
getSharedPreferences("APPNAME",MODE_PRIVATE);
   SharedPreferences.Editor edit = prefs.edit();
   edit.putInt("Last Score", 100);
   edit.commit(); }
```

152. What are different types of logcat methods?

Log.v(): verbose message
Log.d(): debug message
Log.i(): information message
Log.w(): warning message
Log.e(): error message

153. What are Adapter Views?

Adapter Views are Views that lists data using adapters.

ListView: Vertical scrolling list.
Spinner: Drop-down box
GridView: Two dimensional scrolling grid.
AutoCompleteTextView: List of suggested words .

154. What are adapters?

Adapters are objects which are binded to a list of data.

ArrayAdapter: fits AdapterView to an array.

CursorAdapter: fits AdapterView to a cursor.
SpinnerAdapter: fits Spinner to an array.
ListAdapter: fits AdapterView to a List.

155. How would you create a method that returns the name of a particular country based on location (latitude and longitude)?

```
public String getCountry(Double latitude, Double  longitude){
   String countryName=null;
   try {
      Geocoder geo = new Geocoder(this.getApplicationContext(),
Locale.getDefault());
      List<Address> addresses = geo.getFromLocation(latitude, longitude, 1);
         if (addresses.size() > 0) {
            addresses = geo.getFromLocation(latitude, longitude, 1);
            countryName=addresses.get(0).getCountryName();

         }
         else{
            countryName="Not Found";
         }
   }
   catch(Exception e){
      e.printStackTrace();
      Toast.makeText(this, "No Location Name
Found",Toast.LENGTH_LONG).show();
   }
   return countryName;
}
```

156. What are the steps needed to utilize Google Maps?

Download Google Play Services SDK (or just start with Google Map template in Android Studio).

Create permissions to use the Google Map in AndroidManfest.xml

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission
        android:name="android.permission.ACCESS_NETWORK_STATE"
/>
<uses-permission
        android:name="android.permission.WRITE_EXTERNAL_STORAG
E" />
<uses-permission
android:name="com.google.android.providers.gsf.permission.READ_GSE
RVICES" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Obtain an APIKEY for Google Maps: https://code.google.com/apis/console/?pli=1

Insert the Google Map APIKEY into the AndroidManifest.xml

```
<meta-data
        android:name="com.google.android.maps.v2.API_KEY"
        android:value="<APIKEY>" />
```

157. How would you fill your entire activity screen with Google Map?

Create a fragment layout (activity_main.xml)

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
    <fragment
        android:id="@+id/map"

android:name="com.google.android.gms.maps.SupportMapFragment"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
```

```
</RelativeLayout>
```

Create the java code to run the google map within the fragment

```
public class MainActivity extends FragmentActivity {
   static final LatLng Pos = new LatLng(40 , -79);
   private GoogleMap googleMap;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      try {
         if (googleMap == null) {
            googleMap = ((MapFragment) getFragmentManager().
                  findFragmentById(R.id.map)).getMap();
         }
         googleMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
         googleMap.setMyLocationEnabled(true);
         googleMap.setTrafficEnabled(true);
         googleMap.setIndoorEnabled(true);
         googleMap.setBuildingsEnabled(true);
         googleMap.getUiSettings().setZoomControlsEnabled(true);
         Marker marker = googleMap.addMarker(new MarkerOptions().
               position(Pos).title("Hello"));
      } catch (Exception e) {
         e.printStackTrace(); }  }}
```

158. How would you set the latitude and longitude for Google Map with fragment (or xml) layout?

You set it with cameraTargetLat and cameraTargetLng attributes.

```
<fragment xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   xmlns:map="http://schemas.android.com/apk/res-auto"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:id="@+id/map"
   tools:context=".MapsActivity"
```

```
    android:name="com.google.android.gms.maps.SupportMapFragment"
  map:cameraTargetLat="0"
  map:cameraTargetLng="0"
    map:cameraZoom="17"
    />
```

159. How would you set the latitude and longitude for Google Map with java code?

```
 GoogleMap mMap = ((MapFragment)
getFragmentManager().findFragmentById(R.id.map)).getMap();
  LatLng LL=new LatLng(34,-118);
  mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(LL,16));
```

160. How would you get a map of satellite, terrain, hybrid, and normal image with fragment (or xml) layout?

```
map:mapType=”satellite”
map:mapType=”terrain”
map:mapType=”hybrid”
map:mapType=”normal”
```

161. How would you get a map of satellite, terrain, hybrid, and normal map image with java code?

```
GoogleMap mMap = ((MapFragment)
getFragmentManager().findFragmentById(R.id.map)) .getMap();
        mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL);
        mMap.setMapType(GoogleMap.MAP_TYPE_SATELLITE);
        mMap.setMapType(GoogleMap.MAP_TYPE_TERRAIN);
        mMap.setMapType(GoogleMap.MAP_TYPE_HYBRID);
```

162. How do you set markers on Google Map?

```
  GoogleMap mMap = ((MapFragment)
  getFragmentManager().findFragmentById(R.id.map)).getMap();
```

```
static final LatLng PERTH = new LatLng(-31.90, 115.86);
Marker perth = mMap.addMarker(new MarkerOptions()
                    .position(PERTH)
                    .anchor(0.5,0.5)
                    .rotation(90.0));
```

163. How would you set markers on Google Map and center it on the screen?

```
private LatLng latlang= new LatLng(37.5667, 126.9781);
private GoogleMap mMap;
        mMap.addMarker(new
MarkerOptions().position(latlang).title("Marker"));
        CameraPosition cameraPosition = new CameraPosition.Builder()
          .target(latlang)       // Center Set
          .zoom(11.0f)           // Zoom
          .tilt(30)              // Tilt of the camera to 30 degrees
        .build();                // Creates a CameraPosition from the
builder
mMap.animateCamera(CameraUpdateFactory.newCameraPosition(camera
Position));
```

164. How would you create a listener for long click on a Google Map screen?

```
GoogleMap mMap;
mMap.setOnMapLongClickListener(new
GoogleMap.OnMapLongClickListener() {
        @Override
        public void onMapLongClick(LatLng point) {
        }
    });
```

165. How would you create a listener and method for dragging markers on Google Map?

```
GoogleMap mMap;
```

```
            mMap.setOnMarkerDragListener(new
GoogleMap.OnMarkerDragListener() {
            @Override
              public void onMarkerDragStart(Marker marker) {
                    marker.showInfoWindow();
                  }
                  @Override
                 public void onMarkerDrag(Marker marker) {
               CameraPosition cameraPosition = new CameraPosition.Builder()
                     .target(marker.getPosition()) // Center Set
                           .zoom(14)              // Zoom
                           .tilt(10)             // Tilt of the camera to 30 degrees
                           .build();              // Creates a CameraPosition from
the builder

mMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPos
ition));
                        }
          @Override
            public void onMarkerDragEnd(Marker marker) {
              marker.hideInfoWindow();
           }
       });
```

166. What are the 4 java classes related to sensors?

SensorManager: You can use this class to create an instance of the sensor service. This class provides various methods for accessing and listing sensors, registering and unregistering sensor event listeners, and acquiring orientation information. This class also provides several sensor constants that are used to report sensor accuracy, set data acquisition rates, and calibrate sensors.

Sensor: You can use this class to create an instance of a specific sensor. This class provides various methods that let you determine a sensor's capabilities.

SensorEvent: The system uses this class to create a sensor event object, which provides information about a sensor event. A sensor event object includes the following information: the raw sensor data, the type of sensor that generated the event, the accuracy of the data, and the timestamp for the event.

SensorEventListener: You can use this interface to create two callback methods that receive notifications (sensor events) when sensor values change or when sensor accuracy changes.

Source: http://developer.android.com/guide/topics/sensors/sensors_overview.html

167. What the 4 different settings for GPS location settings?

a. Settings.Secure.LOCATION_MODE_SENSORS_ONLY
    i. Network Location Provider disabled, but GPS and other sensors enabled.
b. Settings.Secure.LOCATION_MODE_BATTERY_SAVING
    i. Reduced power usage, such as limiting the number of GPS updates per hour.
c. Settings.Secure.LOCATION_MODE_HIGH_ACCURACY:
    i. Best-effort location computation allowed.
d. Settings.Secure.LOCATION_MODE_OFF
    i. Location mode is off.

168. How would you launch Google GPS Directional Map from an activity?

```
String url =
"google.navigation:q="+TargetLatittude+","+TargetLongitude;
    Uri gmmIntentUri = Uri.parse(url);
    Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
```

```
        mapIntent.setPackage("com.google.android.apps.maps");
        startActivity(mapIntent);
```

169. How would you show YouTube videos on your Android app?

Download the Google Android Player API for youtube (currently located@ https://developers.google.com/youtube/android/player/downloads/). Copy the YouTubeAndroidPLayerApi.jar to the libs folder of the Android Studio project. Go to the YouTube Api Client for java page (currently located @https://developers.google.com/api-client-library/java/apis/youtube/v3). Add to the AndroidManifest.xml file with the following:

```
<uses-permission android:name="android.permission.INTERNET"/>
```

Create the following in the layout xml file

```
<com.google.android.youtube.player.YouTubePlayerView
        android:id="@+id/youtube_view"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>
```

Create the following in the activity java file

```
public class MainActivity extends YouTubeBaseActivity implements
YouTubePlayer.OnInitializedListener {
private static final int RECOVERY_DIALOG_REQUEST = 10;
public static final String API_KEY =
"AIzaSyasdfsafsafsafsafsafsafsafsadfsdfdsafsafasdf";
private String VIDEO_ID = "mcC5XGBDSys";

@Override
```

```
   protected void onCreate(Bundle savedInstanceState) {
     super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_garso_su_gil);

   YouTubePlayerView youTubeView = (YouTubePlayerView)
findViewById(R.id.youtube_view);
   youTubeView.initialize(API_KEY, this);
   }

   @Override
   public void onInitializationFailure(YouTubePlayer.Provider provider,
   YouTubeInitializationResult errorReason) {
      if (errorReason.isUserRecoverableError()) {
         errorReason.getErrorDialog(this,
RECOVERY_DIALOG_REQUEST).show();
      }
     else {
   String errorMessage = String.format("YouTube Error (%1$s)",
         errorReason.toString());
         Toast.makeText(this, errorMessage,
Toast.LENGTH_LONG).show();
      }
   }

  @Override
   public void onInitializationSuccess(YouTubePlayer.Provider provider,
YouTubePlayer player, Boolean
   wasRestored) {
    if (!wasRestored) {
         player.cueVideo(VIDEO_ID);
      }
   }

   @Override
   protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
   if (requestCode == RECOVERY_DIALOG_REQUEST) {
            getYouTubePlayerProvider().initialize(API_KEY, this); }
          }
  protected YouTubePlayer.Provider getYouTubePlayerProvider() {
  return (YouTubePlayerView) findViewById(R.id.youtube_view);
   }}
```

170. How would check if there is a cellular network connectivity?

Add permissions to Android Manifest

```
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission
android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.INTERNET"/>
```

Create a method to check if there is a network connection

```
    protected boolean isOnline() {
    ConnectivityManager cm=
(ConnectivityManager)getSystemService(Context.CONNECTIVITY_SERVICE);

    NetworkInfo netinfo=cm.getActiveNetworkInfo();
    if (netinfo!=null&&netinfo.isConnectedOrConnecting()){
    return true;
    }
    else {return false;}
    }
```

171. What are some of the most commonly used animation methods?

AnimationUtils.makeInAnimation(Context c, boolean fromLeft): Slides animation object from either left or right.

AnimationUtils.makeInChildBottomAnimation(Context c): Slides animation object from up or bottom.

AnimationUtils.makeOutAnimation(Context c, boolean toRight): Slides and fades animation object either left or right.

172. How would you animate a View object to slide and fade from left to right?

```
View viewtoAnimate=findViewById(R.id.theView);
   Animation out= AnimationUtils.makeOutAnimation(this, true);
        viewtoAnimate.startAnimation(out);
   viewtoAnimate.setVisibility(View.INVISIBLE);
```

173. Tell me about 3 different types of animations

a. Property Animation
   i. Introduced in Android 3.0 (API level 11), the property animation system lets you animate properties of any object, including ones that are not rendered to the screen. The system is extensible and lets you animate properties of custom types as well.
b. View Animation
   i. View Animation is the older system and can only be used for Views. It is relatively easy to setup and offers enough capabilities to meet many application's needs.
c. Drawable Animation
   i. Drawable animation involves displaying Drawable resources one after another, like a roll of film. This method of animation is useful if you want to animate things that are easier to represent with Drawable resources, such as a progression of bitmaps.

Source: http://developer.android.com/guide/topics/graphics/overview.html

174. What are the 3 main subclasses of the Animator class in property animation?

ValueAnimator: The main timing engine for property animation that also computes the values for the property to be animated. It has all of the core functionality that calculates animation values and contains the timing details of each animation, information about whether an animation repeats, listeners that receive update events, and the ability to set custom types to evaluate.

ObjectAnimator: A subclass of ValueAnimator that allows you to set a target object and object property to animate.

AnimatorSet: Provides a mechanism to group animations together so that they run in relation to one another. You can set animations to play together, sequentially, or after a specified delay.

175. What are some of the XML attributes of the Property Animation?

The XML file that defines View Animation is stored in /res/animator and the attributes are:

i. <set>
    1. A container that holds other animation elements (<objectAnimator>, <valueAnimator>, or other<set> elements). Represents an AnimatorSet.
ii. <objectAnimator>
    1. Animates a specific property of an object over a specific amount of time. Represents an ObjectAnimator.
iii. <animator>
    1. Performs an animation over a specified amount of time. Represents a ValueAnimator.

176. What are some of the XML attributes of the View Animation?

The XML file that defines View Animation is stored in /res/anim and the attributes are:

i. <set>
    1. A container that holds other animation elements (<alpha>, <scale>, <translate>, <rotate>) or other<set> elements. Represents an AnimationSet.
ii. <alpha>
    1. A fade-in or fade-out animation. Represents an AlphaAnimation.
iii. <scale>
    1. A resizing animation. You can specify the center point of the image from which it grows outward (or inward) by specifying pivotX and pivotY. For example, if these values are 0, 0 (top-left corner), all growth will be down and to the right. Represents a ScaleAnimation.
iv. <translate>
    1. A vertical and/or horizontal motion. Supports the following attributes in any of the following three formats: values from -100 to 100 ending with "%", indicating a percentage relative to itself; values from -100 to 100 ending in "%p", indicating a percentage relative to its parent; a float value with no suffix, indicating an absolute value. Represents a TranslateAnimation.
v. <rotate>
    1. A rotation animation. Represents a RotateAnimation.

177. What is an interpolator?

An interpolator is an animation modifier defined in XML that affects the rate of change in an animation. This allows your existing animation effects to be accelerated, decelerated, repeated, bounced, etc. An interpolator is applied to an animation element with the android:interpolator attribute, the value of which is a reference to an interpolator resource. The XML is located in /res/anim.

178. What is overridePendingTransition(R.id.exitAnimation, R.id.enterAnimation)?

This animation method is typically used in intents and it creates two animation. The first animation (exitAnimation) is the View Animation used to exit out of an activity. The second animation (enterAnimation) is the View Animation used to enter into an Activity.

179. What is the difference between property animation and view animation?

The view animation system provides the capability to only animate View objects, so if you wanted to animate non-View objects, you have to implement your own code to do so. The view animation system is also constrained in the fact that it only exposes a few aspects of a View object to animate, such as the scaling and rotation of a View but not the background color, for instance.

Another disadvantage of the view animation system is that it only modified where the View was drawn, and not the actual View itself. For instance, if you animated a button to

move across the screen, the button draws correctly, but the actual location where you can click the button does not change, so you have to implement your own logic to handle this.

With the property animation system, these constraints are completely removed, and you can animate any property of any object (Views and non-Views) and the object itself is actually modified. The property animation system is also more robust in the way it carries out animation. At a high level, you assign animators to the properties that you want to animate, such as color, position, or size and can define aspects of the animation such as interpolation and synchronization of multiple animators.

The view animation system, however, takes less time to setup and requires less code to write. If view animation accomplishes everything that you need to do, or if your existing code already works the way you want, there is no need to use the property animation system. It also might make sense to use both animation systems for different situations if the use case arises.

180. How would you create an Activity that is sensitive to the user touch gestures?

You can use OnTouchListener interface which only has a single onTouch() method

```
public class TouchTest extends Activity implements
OnTouchListener {
 TextView textView;
 public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
     setContentView(this);}
 public boolean onTouch(View v, MotionEvent event) {
    switch (event.getAction()) {
    case MotionEvent.ACTION_DOWN:
    Toast.makeText(getApplicationContext(), "Action Down",
Toast.LENGTH_LONG).show();
```

```
                break;
               }
                return true; }}
```

181. What is a class?

Class is an abstraction of group of objects typically represented by methods and properties that have a common logical relationship.

182. What is an instantiation?

It is a process in which an object inherits a class and is created in memory.

183. What is the difference between constructors and regular methods?

Constructors must have the same name as the class and it cannot return a value. By default, java will create a default contractor with empty argument in a class without a constructor.

184. What is polymorphism?

The ability for multiple objects to be instantiated using the same class.

185. What is inheritance?

The ability for an object to use the methods and properties of the parent class.

186. What is encapsulation?

Encapsulation is a bundling of logically similar objects into a single logical unit such as a class.

187. What is abstraction?

Abstraction is a technique for managing complexity of computer systems. It works by establishing a level of complexity on which a person interacts with the system, suppressing the more complex details below the current level.

188. What is a difference between an object and a class?

Object is an instance of a class. Class is a mere blueprint in which other objects can be created.

189. What is method overloading?

Method overloading occurs when two or more methods within the same class have the same name but different parameters.

190. What is method overriding?

Method overriding occurs when the child class has the same method name, return type, and parameters as the parent class yet has a different definition (code within the method).

191. What does ‘final” mean when applied to method, class, or variable?

Final method: cannot be overridden by derived class.
Final class: prevents polymorphism.
Final variable: the variable’s value cannot be changed.

192. When does the return statement not return?

The return statement normally exits out of the method that is being executed (including loops). The exception is within the try-catch-finally block in which the finally block will execute even if a return statement was executed within either try or catch block.

193. Is multiple inheritance allowed in Java?

No. Multiple inheritance is the ability for a class to inherit from multiple parent classes.

194. What is an abstract method and class?

Abstract method is a method without a body of code. Abstract class is a class with an abstract method.

195. What is the difference between “==”and equals()?

“==” checks if an objects refers the same place in the memory. equals() compares the values of the two objects.

196. Reverse the elements within array

```
if (inputString==null) inputString="Type String Here";
char[] c=inputString.toCharArray();
//reverse the array of char[]
int N=inputString.length();
for (int i=0; i<N/2; i++) {
     char temp=c[i]; c[i]=c[N-1-i]; c[N-i-1]=temp;
}
String Output=String.valueOf(c);
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

197. Randomly shuffle elements within array?

```
if (input!=null) inputString=input.getText().toString().replaceAll("\\s","");
char[] c=inputString.toCharArray();
int N=inputString.length();
for (int i=0; i<N/2; i++) {
   int randomInteger=i+(int)(Math.random()*(N-i));
   char temp=c[i]; c[i]=c[randomInteger]; c[randomInteger]=temp;
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

198. Suppose that people enter a room until a pair of people share the birthday. On average, how many people will have to enter the room?

```
boolean[] found=new boolean[365];
int daycount=0, foundcount=0;
while (foundcount <365) {
   int day=(int)(Math.random()*365);
   daycount++;
   if(!found[day]){foundcount++;found[day]=true; }
}
```

```
String Output=String.valueOf(daycount);
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

199. Create a method that determines whether an integer is prime.

```
public static boolean isPrime(int N)
{
   if (N<2) return false;
   for (int i=2; i<=N/i; i++){
      if (N%i==0) return false;
   }
   return true;
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

200. Create a method that takes an integer and outputs its factorial (N!) using recursion.

```
public static long factorial(int N)
{
   if(N==1)return 1;
   return N*factorial(N - 1);
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

201. Create a method that takes n number of disks (1,2,3) and outputs all the moves (left, right) for the Tower of Hanoi.

```
public void HanoiMoves(int n, boolean left)
{
   if(n==0) return;
```

```
    HanoiMoves(n-1, !left);
    if(left) Output+="\n"+String.valueOf(n)+ " left";
    else Output+="\n"+String.valueOf(n)+ " right";
    HanoiMoves(n-1,!left);
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

202. Check whether an array of strings is in alphabetical order.

```
public boolean isSorted(String a)
{
    char[] c=inputString.toCharArray();
    for (int i=1; i<c.length; i++) {
        if (c[i - 1]> (c[i])) return false;

    }
    return true;
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

203. Use an insertion sort to sort a string value.

```
    public String InsertSort(String a)
    {
        char[] c=inputString.toCharArray();
        int N=c.length;
        for (int i=1; i<N; i++)
            for (int j=i; j>0; j--)
                if (c[j-1] >c[j]) {char temp=c[j]; c[j]=c[j-1]; c[j-1]=temp;}
                else break;
          return String.valueOf(c);
    }
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

204. Create a class that creates a Stack (LIFO) with push and pop methods.

```
public class Stack<Item>{
    private Node first;
    private int N;
    private String output=null;
    private class Node{
        Item item;
        Node next;
    }
    public boolean isEmpty() {return first==null;}
    public int size() {return N;}
    public void push(Item item){
        Node oldfirst=first;
        first=new Node();
        first.item=item;
        first.next=oldfirst;
        N++;
    }
    public Item pop(){
        Item item=first.item;
        first=first.next;
        N--;
        return item;
    }
    //creates output in string
    public String getOutput() {
        for (Node x = first; x != null; x = x.next) {
            if (output==null) output=x.item.toString();
            else output+=x.item.toString();
        }
        return output;
    }
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

205. Create a class that creates a Queue (FIFO) with enqueue and dequeue methods.

```
public class Queue<Item>{
```

```
    private Node first;
    private Node last;
       private int N;
    private class Node{
       Item item;
       Node next;
    }
    public boolean isEmpty() {return first==null;}
    public int size(){return N;}
    public void enqueue(Item item){
       Node oldlast=last;
       last=new Node();
       last.item=item;
       last.next=null;
       if (isEmpty()) first=last;
       else oldlast.next=last;
       N++;
    }
    public Item dequeue(){
       Item item=first.item;
       first=first.next;
       N--;
       if (isEmpty()) last=null;
       return item;
    }
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

206. How do you delete a 3rd node in a Queue or Stack?

```
public class Queue<Item>{
    private Node first;
    private Node last;
    private int N;
    private class Node{
       Item item;
       Node next;
    }
    public boolean isEmpty() {return first==null;}
    public int size(){return N;}
    public void enqueue(Item item){
       Node oldlast=last;
       last=new Node();
```

```
            last.item=item;
            last.next=null;
            if (isEmpty()) first=last;
            else oldlast.next=last;
            N++;
        }
        public Item dequeue(){
            Item item=first.item;
            first=first.next;
            N--;
            if (isEmpty()) last=null;
            return item;
        }
        public void delete(int nodeToDelete){
            nodeToDelete=2;
            if(size()<nodeToDelete) return;
            int counter=0;
            for (Node x=first; x!=null; x=x.next){
                if (counter==nodeToDelete-1){x.next=x.next.next;  N--;
break;}
                    counter++;
            }
        }
    }
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

207. How would you insert a node with item "@" the 3rd element of a Queue or Stack?

```
    public class Queue<Item>{
        private Node first;
        private Node last;
        private int N;
        private class Node{
            Item item;
            Node next;
        }
        public boolean isEmpty() {return first==null;}
        public int size(){return N;}
        public void enqueue(Item item){
            Node oldlast=last;
```

```
        last=new Node();
        last.item=item;
        last.next=null;
        if (isEmpty()) first=last;
        else oldlast.next=last;
        N++;
    }
    public Item dequeue(){
        Item item=first.item;
        first=first.next;
        N--;
        if (isEmpty()) last=null;
        return item;
    }
    public void insertNode(Item item, int element){
        if(size()<element) return;
        Node insertingNode=new Node();
        insertingNode.item=item;
        int counter=0;
        for (Node x=first; x!=null; x=x.next){
      if (counter==element){insertingNode.next=x.next;
      x.next=insertingNode;  N++; break;}
            counter++;
        }
    }
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

208. Create a method that takes a string, and sorts the characters in the string, and returns the sorted characters as a string using Selection Sort.

```
public String SelectionSort(String a){
    char[] c=a.toCharArray();
    int N=a.length();
    for (int i=0;i<N;i++){
        int min=i;
        for (int j=i+1; j<N;j++)if (c[j] < c[min]) min=j;
        char t=c[i]; c[i]=c[min]; c[min]=t;
    }
    return String.valueOf(c);
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

209. Create a method that takes a string, and sorts the characters in the string, and returns the sorted characters as a string using Insertion Sort.

```
public String InsertionSort(String a){
    char[] c=a.toCharArray();
    int N=a.length();
    for (int i=1; i<N;i++){
        for (int j=i; j>0&&(c[j]<c[j-1]);j--) {char t=c[j]; c[j]=c[j-1]; c[j-1]=t;}
    }
    return String.valueOf(c);
}
```

210. Create a class that takes a string, and sorts the characters in the string, and returns the sorted characters as a string using Merge Sort.

```
public class MergeSort{
    private char[] aux;
    private char[] a;
    private String returnValue;

public String getSortedString() {return returnValue;}
    public MergeSort(String input){
        a=input.toCharArray();
        aux=new char[a.length];
        sort(a, 0, a.length - 1);
        returnValue=String.valueOf(a);
    }
    public void sort(char[] a, int lo, int hi){
        if (hi<=lo) return;
            int mid=lo+(hi-lo)/2;
          sort(a,lo,mid);
          sort(a, mid+1, hi);
          merge(a, lo, mid, hi);
          }
```

```
    public void merge(char[] a, int lo, int mid, int hi){
        int i=lo, j=mid+1;
        for (int k=lo; k<=hi;k++) aux[k]=a[k];
        for (int k=lo; k<=hi; k++) {
            if (i > mid) a[k] = aux[j++];
            else if (j > hi) a[k] = aux[i++];
            else if (aux[j] < aux[i]) a[k] = aux[j++];
            else a[k] = aux[i++];
        }
    }
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

211. Create a class that takes a string, and sorts the characters in the string, and returns the sorted characters as a string using Quick Sort.

```
public class QuickSort {
    private char[] a;
    private String returnValue;
    public String getReturnValue(){return returnValue;}
    private QuickSort(String input) {
        a = input.toCharArray();
        sort(a, 0, a.length - 1);
        returnValue = String.valueOf(a);
    }
    private void sort(char[] a, int lo, int hi){
        if (hi<=lo) return;
        int j=partition(a,lo,hi);
        sort(a,lo,j-1);
        sort(a,j+1,hi);
    }
    private int partition(char[] a, int lo, int hi){
        int i=lo, j=hi+1;
        char v=a[lo];
        while (true){
            while (a[++i]<v) if (i==hi) break;
            while (v<a[--j]) if(j==lo) break;
            if(i>=j) break;
            char t=a[i]; a[i]=a[j]; a[j]=t;
        }
```

```
        char t=a[lo]; a[lo]=a[j]; a[j]=t;
        return j;

    }
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

212. Create a class for Sequential Search (unordered linked list) with get (fetch value based on key) and put (inserts or update value based on key) that returns first 3 characters.

```
    public class SequentialSearch{
        private Node first;
        Character myNull=null;
        private class Node{
            int key;
            char val;
            Node next;
            public Node(int key, char val, Node next){
                this.key=key;
                this.val=val;
                this.next=next;
            }
        }
        public char get(int key){
            for (Node x=first; x!=null; x=x.next){if (key==x.key) return
x.val;}
          return myNull;
        }
        public void put(int key, char val){
            for (Node x=first; x!=null; x=x.next){ if (key==x.key){x.val=val;
return;}}
            first=new Node(key,val, first);
        }
    }
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

213. Create a method to determine if a string has all unique characters

```
public boolean isUnique(String a){
    a=InsertionSort(a);
    char[] c=a.toCharArray();
    int N=c.length;
    for (int x=1; x<N; x++){
      if (c[x-1]==c[x]) return false;
    }
    return true;
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

214. Create a method to determine if two strings are permutations of the other.

```
public boolean isPermutation(String a, String b){
    return (InsertionSort(a).equals(InsertionSort(b)));
}
public String InsertionSort(String a){
    char[] c=a.toCharArray();
    int N=a.length();
    for (int i=1; i<N;i++){
        for (int j=i; j>0&&(c[j]<c[j-1]);j--) {char t=c[j]; c[j]=c[j-1]; c[j-1]=t;}
    }
    return String.valueOf(c);
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

215. Create a method to remove consecutive duplicates from a Stack (LIFO).

```
public class Stack<Item>{
    private Node first;
    private int N;
    private String output=null;
```

```
    private class Node{
          Item item;
          Node next;
      }
      public boolean isEmpty() {return first==null;}
      public int size() {return N;}
      public void push(Item item){
          Node oldfirst=first;
          first=new Node();
          first.item=item;
          first.next=oldfirst;
          N++;
      }
      public Item pop(){
          Item item=first.item;
          first=first.next;
          N--;
          return item;
      }
      public void removeDuplicates() {
          Node x=first;
          while (x.next!=null){
            if(x.item==x.next.item) {x.next=x.next.next; N--;}
            else  if (x.next!=null)x=x.next;
          }
      }

      //creates output in string
      public String getOutput() {
          for (Node x = first; x != null; x = x.next) {
              if (output==null) output=x.item.toString();
              else output+=x.item.toString();
          }
          return output;
      }
  }
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

216. Create a method that returns the maximum value from a Stack (LIFO).

```
public class Stack<Item>{
```

```
    private Node first;
    private int N;
    private String output=null;
    private class Node{
        Item item;
        Node next;
    }
    public boolean isEmpty() {return first==null;}
    public int size() {return N;}
    public void push(Item item){
        Node oldfirst=first;
        first=new Node();
        first.item=item;
        first.next=oldfirst;
        N++;
    }
    public Item pop(){
        Item item=first.item;
        first=first.next;
        N--;
        return item;
    }
    public Item getMaximum(){
        Item maxchar=first.item;
        for (Node x=first; x!=null; x=x.next){
            if (x.item.toString().compareTo(maxchar.toString()) >0)
maxchar=x.item;
        }
        return maxchar;
    }
```

Source: Sedgewick, Robert. Algorithms (4th Edition).

217. Create a method that returns the minimum value from a Queue (FIFO).

```
public class Queue<Item>{
    private Node first;
    private Node last;
    private int N;
    private class Node{
        Item item;
```

```
        Node next;
    }
    public boolean isEmpty() {return first==null;}
    public int size(){return N;}
    public void enqueue(Item item){
        Node oldlast=last;
        last=new Node();
        last.item=item;
        last.next=null;
        if (isEmpty()) first=last;
        else oldlast.next=last;
        N++;
    }
    public Item dequeue(){
        Item item=first.item;
        first=first.next;
        N--;
        if (isEmpty()) last=null;
        return item;
    }
    public Item getMinimum(){
        Item minchar=first.item;
        for(Node x=first; x!=null; x=x.next){
            if (x.item.toString().compareTo(minchar.toString())<0)
minchar=x.item;
        }
        return minchar;
    }
}
```

Source: Sedgewick, Robert. Algorithms (4th Edition).