1. a) Comment every line in the following code

**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);  
  
 }  
}

b) how this code will be linked with the controls in the following xml

<LinearLayout 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:orientation="horizontal">

<EditText android:id="@+id/edit\_message"

android:layout\_weight="1"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:hint="@string/edit\_message" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/button\_send" />

</LinearLayout>

c) How do you comment these two lines:

android:hint="@string/edit\_message"

android:text="@string/button\_send"

1. How do we make two Activities to communicate in this way: we need to send some info, composed by a double and a string from ActivityA to ActivityB. Write the code lines that we use in this case in both activities:

**public class** ActivityA **extends** AppCompatActivity {

double numberd;

String aMessage;

@Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_main***);

// here we send the values into ActivityB using an intent

}

. . .  
}

public class ActivityB extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_display\_message);

// Get the message from the ActivityA

}

......

}

1. a) What is an *Instance state* and what does it happen related to Instance state, when we interact with an activity and press the Back button or if the finish() method of an activity is called.

b) what happens when we interact with an activity and press Home button.

1. Comment the code lines below and explain where they are placed. For the first Activity use the name MainActivity. Complete the method onActivityResult with some code instead of . . .

static final int PICK\_REQUEST = 1;// The request code

Intent i = new Intent(this, SecondActivity.class);

startActivityForResult(i, PICK\_REQUEST);

Intent returnIntent = new Intent();

returnIntent.putExtra("result",result);

setResult(Activity.RESULT\_OK,returnIntent);

Intent returnIntent = new Intent();

setResult(Activity.RESULT\_CANCELED, returnIntent);

finish();

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

// Check which request it is that we're responding to

if (requestCode == 1) {

if(resultCode == Activity.RESULT\_OK){

. . .

. . .

}

if (resultCode == Activity.RESULT\_CANCELED) {

. . .

}

}

}

1. There are different syntaxes that we can use in an Activity to create and manage a list.
2. Our activity extends **ListActivity** class in this case we have the syntax:

ArrayAdapter<String> mAdapter = **new** ArrayAdapter<String>(**this**,  
 R.layout.***activity\_list\_view\_layout***, ***STATES***);  
setListAdapter(mAdapter);  
  
AdapterView.OnItemClickListener mListener = **new** AdapterView.OnItemClickListener() {  
 **public void** onItemClick(AdapterView<?> parent, View view,  
 **int** position, **long** id) {

. . .  
 }  
};  
ListView listView = getListView();  
listView.setOnItemClickListener(mListener);

Explain this case, commenting each line

1. Our activity extends Activity class then what differences do we have in the syntax.
2. Describe the steps to setup a Toolbar
3. What does a Fragment represent in an Android application; if we have the following .xml layout for the MainActivity

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="fill\_parent"  
 android:orientation="horizontal">  
  
 <fragment  
 android:id="@+id/fragment1"  
 android:name="com.example.user.fragmentfirstexample.FirstFragment"  
 android:layout\_weight="1"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
 />  
  
 <fragment  
 android:id="@+id/fragment2"  
 android:name="com.example.user.fragmentfirstexample.SecondFragment"  
 android:layout\_weight="2"  
 android:layout\_width="0dp"  
 android:layout\_height="match\_parent"  
  
 />  
</LinearLayout>

1. What kind of fragments do we have in this case
2. How do we get the reference of one fragment from the other one through the Activity when we use for example the following button. This is part of the Fragment1 code.

**final** Button button =  
 (Button) view.findViewById(R.id.***button1***);

button.setOnClickListener(**new** View.OnClickListener() {   
 **public void** onClick(View v) {  
 buttonClicked(v);  
  
 }});

1. How do we manage dynamically two fragments, FirstFragment and SecondFragment. We want to activate the SecondFragment from the FirstFragment when we click a button in the FirstFragment. In FirstFragment we use the following method of the MainActivity to call the SecondFragment. And we would like to send in the SecondFragment a message (aMessage)

@Override  
**public void** respondToFragment1(**String aMessage**){  
 . . .

secondFragment.setArguments(bundle);  
 FragmentTransaction transaction =

getSupportFragmentManager().beginTransaction();  
 transaction.replace(R.id.***fragment\_container***, fragment);  
 transaction.addToBackStack(**null**);  
 transaction.commit();

}

Describe all the steps that we have to do in FirstActivity and MainActivity for this communication and complete the code.