## W228/307C



### DUBLIN INSTITUTE OF TECHNOLOGY

**BSc. (Honours) Degree in Computer Science**

# Year 3

**WINTER EXAMINATIONS 2016/17**

### MOBILE SOFTWARE DEVELOPMENT CMPU3026

DR. SUSAN MCKEEVER DR. DEIRDRE LILLIS MR KEVIN FOLEY

FRIDAY 6TH JANUARY 1P.M. – 3 P.M. DURATION: 2 HOURS

**INSTRUCTIONS TO CANDIDATES**

QUESTION 1 IS COMPULSORY.

ANSWER QUESTION 1 AND TWO OF THE REMAINING THREE QUESTIONS

QUESTION 1 CARRIES 50 MARKS. ALL OTHER QUESTIONS CARRY 25 MARKS EACH.

|  |  |  |
| --- | --- | --- |
| **Q1.** | **(a)** | Examine the code in Figure 1 Sample Android class and answer the following questions:   1. What is the super class of *MyListAdapter* class? (2 marks)   ArrayAdapter is the super class of MyListAdapter class.   1. How many parameters does the constructor for the MyListAdapter class require? (2 marks)   4 parameters.   1. Take any example of *casting* shown in the code and explain exactly *why* it is required in that example. (3 marks)   ?   1. When developing this class, if you remove the last line of code “return row”, what error will this cause? (3 marks)   If we remove the return row, then we can’t return the actual view.   1. What is the benefit of declaring the attributes “private” in this class? (2 marks)   Encapsulation data and more control on the member data variables.   1. The method setImageResource() is used. Explain what class this method belongs to, and explain your reason. (2 marks)   The class is ImageView, because it use to sets a drawable as the content of this ImageView.   1. Write out the java code for instantiating an *object* of this MyListAdapter class.   You do not need to write any separate array declarations in your answer (2 marks)  setListAdapter(new MyListAdapter(MainActivity.this,R.layout.row,array))   1. What java language feature is being used in the <string> in the class declaration, and what is the purpose of this feature? (4 marks)   ? |
|  |  | (20 marks) |
|  | **(b)** | Explain the purpose of *Interfaces* in java, using listener interfaces in Android event programming to support your answer.  A way to guaranteeing behaviour accurse a set of unrelated classes, if a class implements an interface, can invoke the i/f method safely. And some interfaces are useful for “tagging” classes.  (behaviour that a class “signs” up) for example, in the listener interfaces, assigning that listener to the widget that takes the user action.  (10 marks) |
|  | **(c)** | Explain how Android lifecycle methods onPause() and onResume() can be used to improve the efficiency of location tracking functionality.  Because location updates are use much power, and each provider (GPS, WIFI, Cell based location) are speparately absorbing battery life. Use onPause and onResume to control location update will improve efficiency.  (10 marks) |
|  | **(d)** | A new Android app is to be developed that will allow the general public to report problems to their local council, such as reporting road potholes or broken street lamps. The council will update the status of each problem as it addresses it. The public will be able to see the status of the problem they reported. They will also be able to see the general status of each type of problem that any other users have reported, so that they can have an overall view of how fast the council is fixing problems, and how many problems are being reported.  Explain what choices you would suggest for *persistent data storage* for this app. Clearly explain the reasons for your choices.  Remote cloud database  Security of data  Formal database  Google Fusion  We can store the problem on the remote server. Then update on the local app.  (10 marks) |

|  |  |  |
| --- | --- | --- |
|  |  | public class MyListAdapter extends ArrayAdapter<String> { |
| private String[] names; |
| private String[] desc; |
| private Integer[] imageid; |
| private Activity context; |
| public MyListAdapter(Activity context, String[] names, String[] desc, |
| Integer[] imageid) |
| { |
| super(context, R.layout.list\_layout, names); |
| this.context = context; |
| this.names = names; |
| this.desc = desc; |
| this.imageid = imageid; |
| } |
| @Override |
| public View getView(int position, View convertView, ViewGroup |
| parent) |
| { |
| LayoutInflater inflater = context.getLayoutInflater(); |
| View row = inflater.inflate(R.layout.list\_layout, null, true); |
| TextView textViewName = (TextView) row.findViewById(R.id.textViewName); |
| TextView textViewDesc = (TextView) row.findViewById(R.id.textViewDesc); |
| ImageView image = (ImageView) row.findViewById(R.id.imageView); |
| textViewName.setText(names[position]); |
| textViewDesc.setText(desc[position]); |
| image.setImageResource(imageid[position]); |
| return row; |
| } |
| ] |
| **Figure 1: Sample Android class** |
| **Q2.** | **(a)** | Figure 2 shows a list in Android. Assume that the two pieces of text in each row are separately provided (e.g. “HTML” and “The Powerful Hyper Text Mark Up” are two separate pieces of text). Answer the following:   1. How many layouts are required to implement this list? Explain your answer.   2 xml layout,  one simple row layout: to display a simple one column list of text items. 1 xml screen layout includes<listview>.  One custom row layouts: to use your own row layout. 1 xml.  (4 marks)   1. Write the XML code for the *row layout* required for this list.   Simple layout:  <ListView xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent"  android:background="@color/colorAccent"  android:cacheColorHint="#00000000"  android:id="@android:id/list">  </ListView>  Custom layout:  <ImageView  android:id="@+id/imageview"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:src="@drawable/image"/>  <linerlayout  android:layout\_width="fill\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical">  <TextView  android:id="@+id/textviewname"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"/>  <TextView  android:id="@+id/textviewdesc"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"/>  </linerlayout>  (6 marks)  (Total Q2(a) 10 Marks) |
|  |  |  |

|  |  |
| --- | --- |
|  |  |
|  | **Figure 2: Sample Android List** |
| **(b)** | The code shown in Figure 1 Sample Android Class is the custom adapter class that will be used by a ListActivity to populate this list. Answer the following:   1. When developing the custom ListActivity class (which is not shown here), what code (in your own words) is needed to link the custom ListActivity   class to this adapter? Include any suggestions, with reasons, on *class structure*  in your answer.  ?  (6 marks)   1. The list is to be changed so that an additional piece of text will appear on the right hand side of *each row* of the list (showing the inventor name of the language). Explain the code changes needed to both *XML Layouts(s)* and the *custom adapter MyListAdapter class* to implement this. Assume that the inventor names are available in an extra array passed in to the adapter.   Add new textview in the custom xml. In the MyListAdapter class need add inventor name.  (9 marks)  (Total Q2(b) 15 Marks) |

|  |  |  |
| --- | --- | --- |
| **Q3.** | **(a)** | In mobile apps, delays at the user interface should be avoided. Any long tasks should be pushed to background processing if possible. *Explain* how Android supports background processing, and *how* this is implemented. Also include a scenario *example* to support your explanation.  In android consider to use of threads. The subclass call AsyncTask.  Must override the doInBackGround() method, which runs on separate thread.  Eg. When getWriteableDatabase() to connect to the database, it takes long time, so can pushed to background processing.  Code:  private class MyLongTask extends AsyncTask  {  @Override  protected Object doInBackground(final Object... objects)  {  try  {  openHelperRef.getWritableDatabase();  }  catch (etc)  }}  (10 Marks) |
|  | **(b)** | Model-View-Controller (MVC) is a popular software architecture. *Explain* the following:   1. The general principles of MVC and its advantages; (5 marks) 2. The extent to which the *Android* framework supports MVC for app development. (5 marks)   (i)  based on the idea that the logic of an application should be separated from its presentation. Put simply, I would say that MVC is simply a better way of separating the logic of your application from the display. • Model = classes that deal with storing and using data  - If data structure(s) or rules change. Just change these classes  – Can have multiple different GUIs using the same model  • View = classes or code that deal with the GUI part  If the GUI layout changes, don’t (necessarily) need to change the model or control classes  • Controller = (non graphical) classes that bridge the View and the Model  Keep separate  - Listener classes that trigger behaviour  - Generally.. Consists of everything that is not the layout or not directly storing/using data  Philosophy  • Keep the parts that may be reused elsewhere, maintainability or changed separate.  (ii)?  Abstracted screen layout: easy to maintain if changed without necessarily changing M/C parts.  Static resources are kept separate from dynamic code.  Dynamic model code.  (10 Marks) |
|  | **(c)** | The practice of embedding static data in dynamic code is best avoided as it can lead to the need to re-test software if static data is changed. *Explain in your own words* how the Android framework helps the developer to avoid this. Use data such as arrays or colour settings as an example.  ?  For colour settings, I define a colour in the resource colour.xml. If I want use this colour, I will use colour name to call this color in the layout xml.  E.g I made a red colour,#FF0000, name is red. If I want a textview a textcolor is red. I can android:textColor="@color/red" to assign the this text is red color.  (5 marks) |
| **Q4.** | **(a)** | A developer wishes to add functionality to a button on an Android app screen, whereby a click on the button results in a particular action. Describe *in your own words three* different ways in Android to implement event programming for a button click.   1. Listeners implemented in the class   Dynamic code is in one place +  Shared listener across widgets = shared callback -  Longer to code -   1. Anonymous listener classes for each widget   Can’t reuse – separate class per widget -  potential performance hit -  Easier to follow code? +   1. Embed event handler method into XML   Simplest to implement +  But  1. Presentation coupled with logic- bad –  2. Changes to method name - > need to remember to refactor the xml –  3. Multiple XML files using a single method can lead to maintenance problems if functionality diverges -  (10 Marks) |
| **(b)** | A mobile app developer wishes to develop a new app with that will work on as many end user devices as possible amongst the general public. Explain the following:   1. The choices available to the developer of *native* versus *hybrid* app and reasons for choosing either.   Native app   * Single platform affinity * Written with platform SDKs * Must be written for each platform * Access to all native APIs * Faster graphics performances * Appstore distribution   Hybrid app.   * Cross-platform affinity * Written with web technologies * Run locally on the devices, supports offline * Access to native APIs * Appstore distribution * (5 marks)  1. How the developer could research the *variety of devices* and *Android versions*   amongst Android users, if Android is chosen as the platform.  ?  (5 marks)   1. *Five* techniques provided by the Android frame to enable apps to run successfully on a variety of screen devices 2. Use Screen compatibility mode 3. Specify Screens supported 4. Provide multiple screen layouts 5. Provide multiple image densities 6. Use Best practices 7. Use fragments (Android 3 onwards)   (5 marks) | |
|  |  | |