Midterm Exam #1

CMSC 436
Programming Handheld Devices
Fall 2011

October 19, 2011

Guidelines

Put your name on each page before starting the exam. Write your answers directly on the exam sheets, using the back of the page as necessary. If you finish with more than 15 minutes left in the class, then bring your exam to the front when you are finished and leave the class as quietly as possible. Otherwise, please stay in your seat until the end.

If you have a question, raise your hand and I will come to you. Note, that I am unlikely to answer general questions however. If you feel an exam question assumes something that is not written, write it down on your exam sheet. Barring some unforeseen error on the exam, however, you shouldn't need to do this at all, so be careful when making assumptions.

Question	Points	Score
1	30	
2	20	
3	25	
4	25	
Total	100	

1. Short answers (30 points). Give very short (1 to 2 sentences for each issue) answers to the following questions. **Longer responses to these questions will not be read.**

   (a) In class we mentioned 4 fundamental components of Android applications. Name them and for each give a 1-sentence explanation of their purpose.

   Answer:

   Activity - Provides a user interface supporting one task a user can do
   Service - Performs background operations and allows communication across applications
   BroadcastReceiver - Listens for and reacts to events
   ContentProvider - Cross-application database

   (b) Assume you have an application that is already running an Activity called Activity1. Activity1 starts another Activity called Activity2. Name one Activity lifecycle method that will be called on Activity1 after this point, but before Activity2 starts.

   Answer:

   onPause() or onStop()

   (c) What is the purpose of the IntentFilter class? Name/describe two pieces of information that can be specified in an IntentFilter.

   Answer:

   IntentFilters specify Intents that a component can receive. Two pieces of IntentFilter data include any two of Label, Icon, Priority, Action, Category, or Data.

(d) Suppose you have an application that is running an Activity called Activity1. Suppose that Activity1 excutes and starts other Activities, but that the user never quits or backs out of the Activity. How many times can Activity1's onCreate() method get called? Explain your answer.

Answer:

Any number of times. Because the sytem can kill Activity1 while it's not running. If this happens and the user later returns to Activity1, onCreate() will be called.

(e) Name two kinds of menus supported by the Android Platform.

Answer:

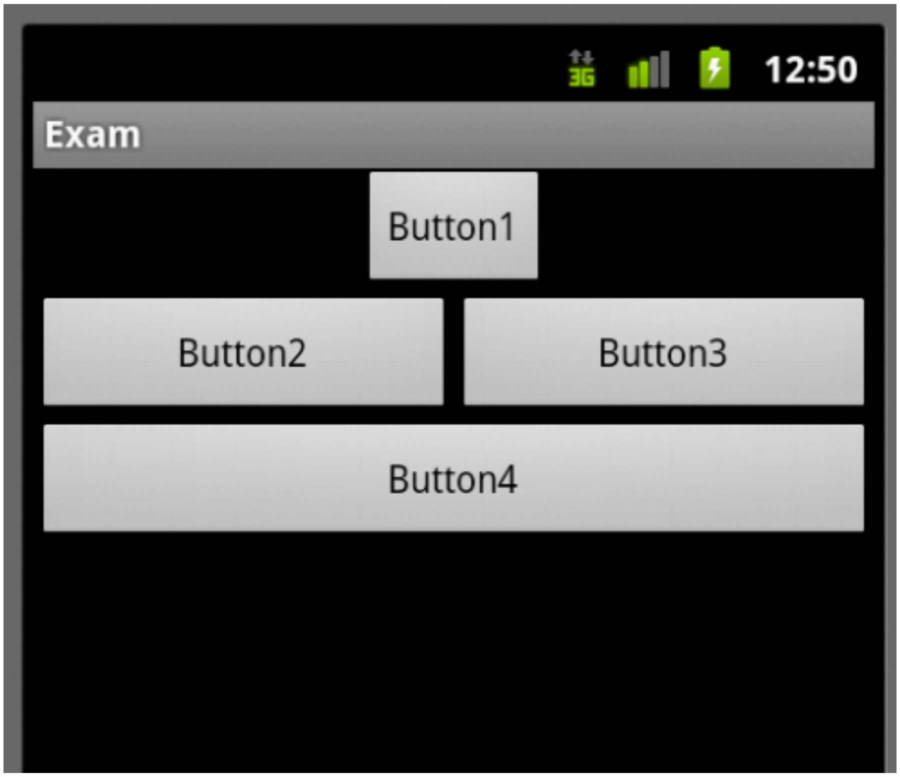
Any two of Options, Context and SubMenus.

(f) True or False: Dynamically-registered BroadcastReceivers will not receive Intents broadcast with Context.sendBroadcast () if the Activity that registered them is paused when state when the Intent is broadcast.

Answer:

False.

2. Layouts. (20 points). Outline the contents of an XML file, called main.xml, that would generate the layout shown below. Write your answer on the next page.



Answer:

```
...
<LinearLayout ... >
  <Button android:text="Button1" android:layout_height="wrap_content"
        android:layout_gravity="center" android:layout_width="wrap_content" ...>
  </Button>
  <LinearLayout android:layout_height="wrap_content"
                android:layout_width="match_parent" ... >
    <Button android:text="Button2" android:layout_height="wrap_content"
            android:layout_gravity="center" android:layout_weight="1"
            android:layout_width="0dp" ... >
    </Button>
    <Button android:text="Button3" android:layout_height="wrap_content"
            android:layout_gravity="center" android:layout_weight="1"
            android:layout_width="0dp" ...>
    </Button>
  </LinearLayout>
    <Button android:text="Button4" android:layout_height="wrap_content"
            android:layout_gravity="center" android:layout_width="fill_parent" ...>
    </Button>
</LinearLayout>
```

3. BroadcastReceivers. (25 points). Assume that there may be multiple 3^{rd}-party applications on your device that register BroadcastReceivers to receive text Tweets in order to process them in various ways (e.g., logging, displaying, etc.). Assume that incoming Tweets are received by a class called RawTweet and are rebroadcast to all BroadcastReceivers registered for an Intent with the action field NEW_TWEET as defined below in the RawTweet class. Assume you control RawTweet and can therefore modify it.

   Suppose you have a new BroadcastReceiver, called TweetScreenerBroadcastReceiver, whose goal is to scan incoming Tweets and to remove any Tweets with offensive language before so they cannot be viewed, processed or stored by any of the 3^{rd}-party applications on your device.

   You are not allowed to modify any of the 3^{rd}-party components. How can you implement the above scenario? Are there any assumptions you need to make to get your application to work properly?

```
public class RawTweet {
  public static String NEW_TWEET = "edu.umd.cs.cmsc436.Midterm.BroadcastReceiver.NEW_TWEET";
  public static String TWEET_TEXT = NEW_TWEET + ".TEXT";

...
  // The text of the incoming Tweet
  String tweetText = ...
  Intent intent = new Intent(NEW_TWEET);
  intent.putExtra(TWEET_TEXT, tweetView.getText().toString());

// FILL IN

// Create IntentFilter with high priority for TweetScreenBroadcastReceiver when
// receiving NEW_TWEET Intent. I did this in the AndroidManifest.xml file. You
// can do this in code as well

...
  sendOrderedBroadcast(intent, null);
...
}
```

```
}
```

over...

```
public class TweetScreenerBroadcastReceiver extends BroadcastReceiver {
  private final String LOGTAG = "ScreenerBroadcastReceiver";
  public void onReceive(Context context, Intent intent) {

    if (intent.getAction().equals(RawTweetActivity.NEW_TWEET)) {
      String text = intent.getStringExtra(RawTweetActivity.TWEET_TEXT);
      if (!ok(text) && isOrderedBroadcast())
        abortBroadcast();
      }
    }



    }

  private boolean ok (String text) {
    // Assume this does the right thing
    }
}
```

4. ListViews. (25 points) Fill in the code below to finish creating a ListActivity that presents a list of “Favorite Colors.” Once the list is visible, users can click on a given color to remove item from the favorite colors list. You can, but don’t have to, fill in code whereever you see // FILL IN.

```
public class ColorListActivity extends ListActivity {
  final ArrayList<String> data = new ArrayList<String>();
  // FILL IN



  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    // R.array.colors is an array of colors (e.g., "red", "white", "blue")
    data.addAll(Arrays.asList(getResources().getStringArray(R.array.colors)));
    // R.layout.row defines a simple TextView
    final ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,R.layout.row, data);
  // FILL IN


    setListAdapter(adapter);


    getListView().setOnItemClickListener(new OnItemClickListener() {
     // parent - parent ListView, view - item TextView, position - item selected, id - N/A
     public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
  // FILL IN

        adapter.remove(adapter.getItem(position));




     }
   });
  }
  // FILL IN







}
```