CS260 Android working with Multiple Activities & Intents

We are going to continue to add functionality to our Calculator application that we started last time.

(a) Add two menu items "About" and "Print Integers" to your activity.

```xml
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
<item
    android:id="@+id/menuAbout"
    android:orderInCategory="0"
    android:showAsAction="never"
    android:title="@string/menuAbout"/>
<item
    android:id="@+id/menuPrintInts"
    android:orderInCategory="1"
    android:showAsAction="never"
    android:title="@string/menuPrintInts"/>
<item
    android:id="@+id/menuSettings"
    android:orderInCategory="100"
    android:showAsAction="never"
    android:title="@string/menuSettings"/>
</menu>
```

1. Notice for ids I'm using menu as a prefix. You will need to modify the menu item id and string name for Settings.
2. Run your modified activity and make sure the menu with About, Print Integers, and Settings shows up.

(b) Create another activity in the current package called AboutActivity.java by right-clicking on the package ->New->Other->Android Activity.

   i.   This will be a blank activity and not a Launcher Activity as we don't want this activity to have an icon and exist with other launchable activities.
   ii. Also notice that Eclipse shows what will be added to the AndroidManifest.xml file. That is, the manifest file allows the new activity to run.
   iii. Simply display a message "Hello About" when this activity is executed.

(c) Process menu selections by adding the following code after the onCreateOptionsMenu in the main activity.

```java
public boolean onOptionsItemSelected (MenuItem item)
{
  Intent intent;

  switch ( item.getItemId ())
  {
    case R.id.menuAbout:
      intent = new Intent (this, AboutActivity.class);
      startActivity (intent);
      break;

    case R.id.menuPrintInts:
      break;

    case R.id.menuSettings:
      break;
  }
  return true; // assume we handled the event
}
```

      i.   Run the main activity and then select the menu item About. This should start the About Activity and display the message "Hello About"

d) Create an Activity PrintInts.java that will initially display a that displays "PrintInts Activity" when the activity is started. You need to go through the same steps as you did with the AboutActivity.

e) How can we create a View in code versus XML? Take a look at the following code and replace the existing PrintIntsActivity class code with this class code. Run the activity and see what the output looks like.

```java
public class PrintIntsActivity extends Activity
{

  private LinearLayout mRoot;
  private TextView mTextView;

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

    LinearLayout.LayoutParams linearContainerParams =
        new LinearLayout.LayoutParams (
            ViewGroup.LayoutParams.MATCH_PARENT,
            ViewGroup.LayoutParams.WRAP_CONTENT,
            0.0f);
```

```
        LinearLayout.LayoutParams linearWidgetParams =
            new LinearLayout.LayoutParams (
                ViewGroup.LayoutParams.MATCH_PARENT,
                ViewGroup.LayoutParams.MATCH_PARENT,
                1.0f);

        mRoot = new LinearLayout (this);
        mRoot.setOrientation (LinearLayout.VERTICAL);
        mRoot.setBackgroundColor (Color.LTGRAY);
        mRoot.setLayoutParams (linearContainerParams);

        mTextView = new TextView (this);
        mTextView.setText ("Balance($)");
        mTextView.setTextColor (Color.RED);
        mTextView.setGravity (Gravity.RIGHT);
        mTextView.setLayoutParams (linearWidgetParams);

        mRoot.addView (mTextView);

        setContentView (mRoot);
    }
```

f) Pass information between the Activities.

In MainActivity

```
     intent = new Intent (this, PrintIntsActivity.class);
     intent.putExtra ("int", 10);
```

In PrintInts Activity

```
    private int mValue;
    mValue = getIntent ().getExtras ().getInt ("int");
```

    i.   Use the debugger and see if mValue is the correctly passed in value.

g) Using the passed in integer, create a table of values 1 to the integer passed in one row per integer. As a starting point, here is a row in a table with two headings.

```
  LinearLayout.LayoutParams linearContainerParams =
      new LinearLayout.LayoutParams (
          ViewGroup.LayoutParams.MATCH_PARENT,
          ViewGroup.LayoutParams.WRAP_CONTENT,
          0.0f);

  LinearLayout.LayoutParams linearWidgetParams =
```

```java
        new LinearLayout.LayoutParams (
            ViewGroup.LayoutParams.MATCH_PARENT,
            ViewGroup.LayoutParams.MATCH_PARENT,
            1.0f);

    TableLayout.LayoutParams tableContainerParams =
        new TableLayout.LayoutParams (
            ViewGroup.LayoutParams.MATCH_PARENT,
            ViewGroup.LayoutParams.WRAP_CONTENT,
            0.0f);

    TableLayout.LayoutParams tableWidgetParams =
        new TableLayout.LayoutParams (
            ViewGroup.LayoutParams.MATCH_PARENT,
            ViewGroup.LayoutParams.MATCH_PARENT,
            1.0f);

    TableRow.LayoutParams rowContainerParams =
        new TableRow.LayoutParams (
            ViewGroup.LayoutParams.MATCH_PARENT,
            ViewGroup.LayoutParams.WRAP_CONTENT,
            0.0f);

    TableRow.LayoutParams rowWidgetParams =
        new TableRow.LayoutParams (
            ViewGroup.LayoutParams.MATCH_PARENT,
            ViewGroup.LayoutParams.MATCH_PARENT,
            1.0f);

    mRoot = new LinearLayout (this);
    mRoot.setOrientation (LinearLayout.VERTICAL);
    mRoot.setBackgroundColor (Color.LTGRAY);
    mRoot.setLayoutParams (linearContainerParams);

    mTableLayout = new TableLayout (this);
    mTableLayout.setOrientation (TableLayout.VERTICAL);
    mTableLayout.setBackgroundColor (Color.BLUE);
    mTableLayout.setLayoutParams (tableContainerParams);
    mRoot.addView (mTableLayout);

    mTableRow = new TableRow (this);
    mTableRow.setOrientation (TableLayout.VERTICAL);
    mTableRow.setBackgroundColor (Color.CYAN);
    mTableRow.setLayoutParams (rowContainerParams);
    mTableLayout.addView (mTableRow);

    mTextView = new TextView (this);
    mTextView.setText ("Balance($)");
    mTextView.setTextColor (Color.RED);
    mTextView.setGravity (Gravity.RIGHT);
    mTextView.setLayoutParams (rowWidgetParams);
    mTableRow.addView (mTextView);

    mTextView = new TextView (this);
    mTextView.setText ("Month");
```

```java
mTextView.setTextColor (Color.RED);
mTextView.setGravity (Gravity.RIGHT);
mTextView.setLayoutParams (rowWidgetParams);
mTableRow.addView (mTextView);

setContentView (mRoot);
```