Lab assignment 2

Due date: Friday, June 3

40 points

Team of 3 to 4 students

Divide the team into two subgroups with two team members per group.

The group works on the lab assignment and can discuss the code with another group. Each group demonstrates the lab to each other. The team then upload the solution from one of the subgroups.

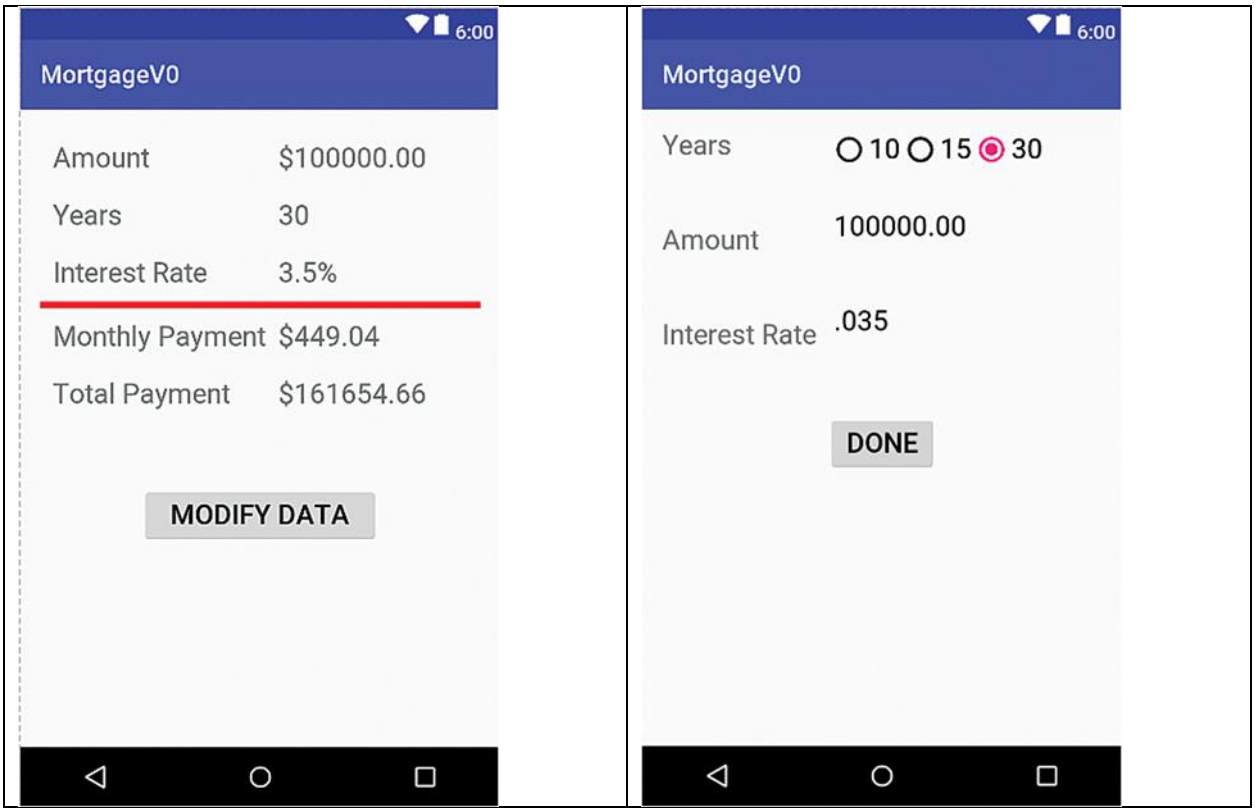
Write the contribution to the lab assignment, the contribution percentage (maximum of 100%) on the comment box for each team member.

All the team members must agree on the percentage. If the team members have 50% and the lab assignment grade is 40/40, they get only 20 points. The instructor will review the paper summarizing and validate the percentage of each team member. If you have problems with team members, please let me know.

Problem

In this lab assignment you will learn how to code several activities, how to go from one to another and back, how to share data between activities, how to set up transitions between them, and how to save the state of an app and retrieve it whenever the user starts the app again (i.e., how to make the data persistent). We build a mortgage calculator app as a vehicle to learn all these concepts.

Design



Model

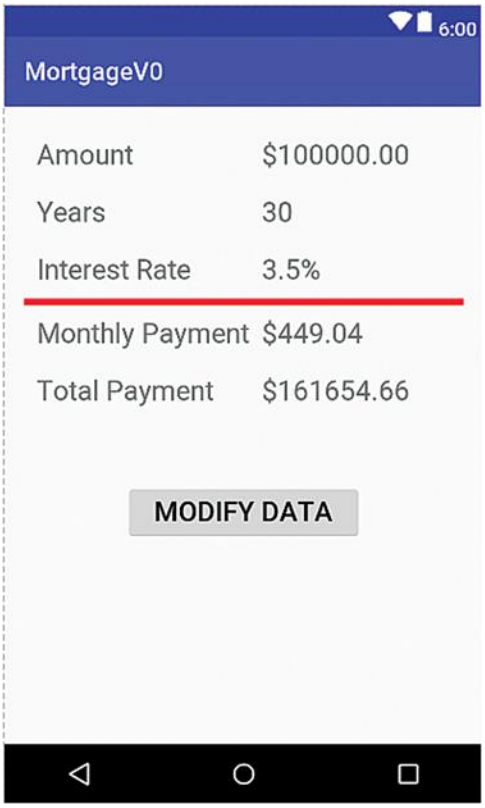
For this app, the Model is simple and is only composed of one class that encapsulates a mortgage calculator, the Mortgage class. It is a regular Java class.

Mortgage class

```
package com.example.mortgagev0;
import java.text.DecimalFormat;

public class Mortgage {
  public final DecimalFormat MONEY
               = new DecimalFormat( "$#,##0.00" );

  private float amount;
  private int years;
  private float rate;

  public Mortgage( ) {
    setAmount( 100000.0f );
    setYears( 30 );
    setRate( 0.035f );
  }

  public void setAmount( float newAmount ) {
    if( newAmount >= 0 )
```

```
      amount = newAmount;
  }
  public void setYears( int newYears ) {
    if( newYears >= 0 )
      years = newYears;
  }

  public void setRate( float newRate ) {
    if( newRate >= 0 )
      rate = newRate;
  }

  public float getAmount( ) {
    return amount;
  }

  public String getFormattedAmount( ) {
    return MONEY.format( amount );
  }

  public int getYears( ) {
    return years;
  }

  public float getRate( ) {
    return rate;
  }

  public float monthlyPayment( ) {
    float mRate = rate / 12;  // monthly interest rate
    double temp = Math.pow( 1/( 1 + mRate ), years * 12 );
    return amount * mRate / ( float ) ( 1 - temp );
  }

  public String formattedMonthlyPayment( ) {
    return MONEY.format( monthlyPayment( ) );
  }

  public float totalPayment( ) {
    return monthlyPayment( ) * years * 12;
  }

  public String formattedTotalPayment( ) {
    return MONEY.format( totalPayment( ) );
  }
}
```

Using a TableLayout for the Front Screen GUI: Mortgage Calculator App, Version 0



activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
   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:layout_margin = "@dimen/activity_margin"
   tools:context="com.jblearning.mortgagev0.MainActivity" >

   <TableRow
      android:layout_width="wrap_content"
      android:layout_height="wrap_content" >
      <TextView
         android:text="@string/label_amount"
         android:padding="10dip" />
      <TextView
         android:id="@+id/amount"
```

```
        android:text="@string/amount" />
    </TableRow>

    <TableRow
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
        <TextView
            android:text="@string/label_years"
            android:padding="10dip" />
        <TextView
            android:id="@+id/years"
            android:text="@string/years" />
    </TableRow>

    <TableRow
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >
        <TextView
            android:text="@string/label_rate"
            android:padding="10dip" />
        <TextView
            android:id="@+id/rate"
            android:text="@string/rate" />
    </TableRow>

    <!-- red line -->
    <View
        android:layout_height="5dip"
        android:background="#FF0000" />

    <TableRow

    </TableRow>

    <TableRow

    </TableRow>

    <TableRow

    </TableRow>

</TableLayout>
```

dimens.xml

```
<resources>
 <!-- Default screen margins, per the Android Design guidelines. -->
 <dimen name="activity_margin">16dp</dimen>
</resources>
```

strings.xml

```
<resources>
   <string name="app_name">MortgageV0</string>
   <string name="label_amount">Amount</string>
   <string name="amount">$100000.00</string>
   <string name="label_years">Years</string>
   <string name="years">30</string>
   <string name="label_rate">Interest Rate</string>
   <string name="rate">3.5%</string>
   <string name="label_monthly_payment">Monthly Payment</string>
   <string name="monthly_payment">$449.04</string>
   <string name="label_total_payment">Total Payment</string>
   <string name="total_payment">$161654.66</string>
   <string name="modify_data">Modify Data</string>

</resources>
```

styles.xml

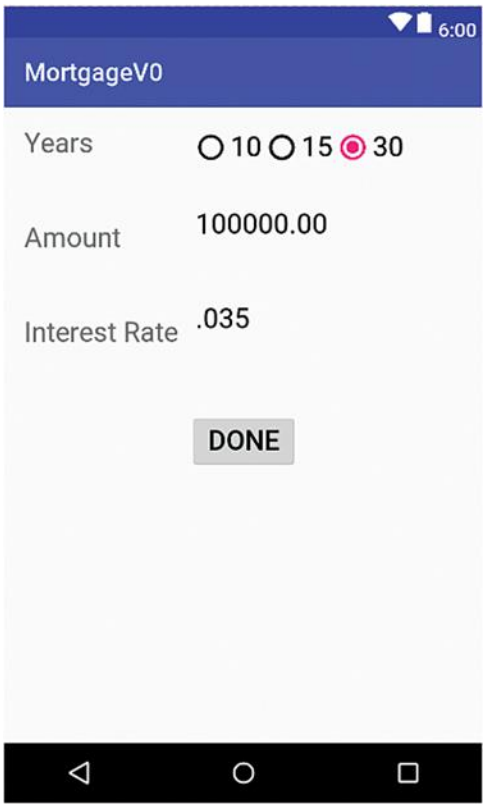
```
<resources>
   <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
      <item name = "android:textSize">22sp</item>
      <item name="colorPrimary">@color/colorPrimary</item>
      <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
      <item name="colorAccent">@color/colorAccent</item>
   </style>
</resources>
```

MainActivity class

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

```
    }
}
```

Using a RelativeLayout for the second screen UI
You need to change the layout to a ConstraintLayout design



Activity_data.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 android:orientation="vertical"
 android:layout_margin="@dimen/activity_margin" >

 <TextView
  android:id="@+id/label_years"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/label_years"/>
```

```
<RadioGroup
  android:layout_toRightOf="@+id/label_years"
  android:layout_width="match_parent"
  android:layout_height="wrap_content"
  android:layout_alignLeft="@+id/data_rate"
  android:orientation="horizontal">

  <RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/ten"
    android:text="@string/ten" />
  <RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/fifteen"
    android:text="@string/fifteen" />
  <RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/thirty"
    android:checked="true"
    android:text="@string/thirty" />

</RadioGroup>

<TextView
  android:id="@+id/label_amount"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@+id/label_years"
  android:layout_marginTop="50dp"
  android:text="@string/label_amount" />

<EditText
  android:id="@+id/data_amount"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignBottom="@+id/label_amount"
  android:layout_alignLeft="@+id/data_rate"
  android:layout_alignParentRight="true"
  android:layout_toRightOf="@+id/label_amount"
  android:text="@string/amountDecimal"
  android:inputType="numberDecimal" />

<TextView
  android:id="@+id/label_rate"
  android:layout_width="wrap_content"
```

```
   android:layout_height="wrap_content"
   android:layout_marginTop="50dp"
   android:layout_below="@+id/label_amount"
   android:text="@string/label_rate" />

  <EditText
   android:id="@+id/data_rate"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_alignBottom="@+id/label_rate"
   android:layout_alignParentRight="true"
   android:layout_toRightOf="@+id/label_rate"
   android:layout_marginLeft="10dp"
   android:text="@string/rateDecimal"
   android:inputType="numberDecimal" />

  <Button
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_centerHorizontal="true"
   android:layout_below="@+id/data_rate"
   android:layout_marginTop="50dp"
   android:onClick="goBack"
   android:text="@string/done" />

</RelativeLayout>
```

Strings.xml (updating)

```
   <string name="ten">10</string>
   <string name="fifteen">15</string>
   <string name="thirty">30</string>
   <string name="amountDecimal">100000.00</string>
   <string name="rateDecimal">.035</string>
   <string name="done">Done</string>
```

MainActivity class (updating)

```
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

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

```
  }
}
```

Connecting Two Activities: Mortgage Calculator App, Version 1

You have now defined and coded the Model and the View part of the app, for which we have two XML files defining two Views. Next, you code the Controller part of the app. You want to be able to navigate back and forth between the two Views that we have created. For this, you need to complete the following steps:

- Add some code in the first Activity class so that we can go to the second View via some user interaction, in this case when the user clicks on the Modify Data button.

- Add a new class, DataActivity, which extends Activity, to manage the second View. Include the code so that we can dismiss that activity and go back to the first View when the user clicks on the Done button.

- Add another activity element (for the second activity) in the AndroidManifest.xml file.

The Intent and Activity classes provide the functionality to start a new activity and go back to a previous activity. The Intent class encapsulates the concept of an operation to be performed. It is typically used to launch a new activity, and can also be used to launch a service.

MainActivity class (updating)

```
public class MainActivity extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }

  public void modifyData( View v ) {
    Intent _____ = _____
    this.startActivity(_____);
  }
}
```

DataActivity class

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
```

```
import android.view.View;

public class DataActivity extends AppCompatActivity {

  public void onCreate( Bundle savedInstanceState ) {
   super.onCreate( savedInstanceState );
   _____________________________________________
  }

  public void goBack_____________________ {
_____________________________________
  }
}
```

AndroidManifest.xml (updating)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest package="com.jblearning.mortgagev1"
        xmlns:android="http://schemas.android.com/apk/res/android">

  <application
     android:allowBackup="true"
     android:icon="@mipmap/ic_launcher"
     android:label="@string/app_name"
     android:supportsRtl="true"
     android:theme="@style/AppTheme">
    <activity android:name=".MainActivity"
       android:screenOrientation="portrait">
     <intent-filter>
       <action android:name="android.intent.action.MAIN"/>

       <category android:name="android.intent.category.LAUNCHER"/>
     </intent-filter>
    </activity>

    ←Define a second activity→
  </application>

</manifest>
```

Sharing Data between Activities: Mortgage Calculator App, Version 2

In Version 2, you add functionality to the Controller in order to have a fully functional app. For this, you need to be able to pass the values input by the user in the second View to the activity managing the first View so that you can compute the monthly and total payments and display them. When you go back to the second View to edit the values againyou need to retrieve and show the most recent values, not the default values.

There are several ways that we can pass data from one activity to another, including:

‣ Pass data using the putExtra methods of the Intent class. Data must be either primitive data types or Strings.

‣ Declare a public static instance of a class of the Model (in this app, the Mortgage class) in one Activity class. That makes that instance globally accessible by any other Activity class.

‣ Rewrite the Mortgage class as a “singleton” class so that all the Activity classes can access and share the same object.

‣ Write data to a file and read it from that file.

‣ Write data to a SQLite database and rea from it.

MainActivity class (updating)

```
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
  public static Mortgage mortgage;

  protected void onCreate( Bundle savedInstanceState ) {
    super.onCreate( savedInstanceState );
    mortgage = new Mortgage( );
    setContentView( R.layout.activity_main );
  }

  public void onStart( ) {
    super.onStart( );
    updateView( );
  }
```

```
 public void updateView( ) {
  TextView amountTV = ( TextView ) findViewById( R.id.amount );
  amountTV.setText( mortgage.getFormattedAmount( ) );
  TextView yearsTV = ( TextView ) findViewById( R.id.years );
  yearsTV.setText( "" + mortgage.getYears( ) );

  ……..
 }

 public void modifyData( View v ) {
  _____
  _____
 }
}
```

DataActivity class (updating)

```
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.EditText;
import android.widget.RadioButton;

public class DataActivity extends AppCompatActivity {
 public void onCreate( Bundle savedInstanceState ) {
  super.onCreate( savedInstanceState );
  _____
  _____
 }

 public void updateView( ) {
  Mortgage mortgage = MainActivity.mortgage;
  if( mortgage.getYears( ) == 10 ) {
   RadioButton rb10 = ( RadioButton ) findViewById( R.id.ten );
   rb10.setChecked( true );
  } else if ……
   ……..
  } // else do nothing (default is 30)

  EditText amountET = ( EditText ) findViewById( R.id.data_amount );
  amountET.setText( "" + mortgage.getAmount( ) );

  ……
 }

 public void updateMortgageObject( ) {
  Mortgage mortgage = MainActivity.mortgage;
```

```
   RadioButton rb10 = ( RadioButton ) findViewById( R.id.ten );
   RadioButton rb15 = ( RadioButton ) findViewById( R.id.fifteen );
   int years = 30;
   if( rb10.isChecked( ) )
     years = 10;
   else if ……..
   ……
   mortgage.setYears( years );
   EditText amountET = ( EditText ) findViewById( R.id.data_amount );
   String amountString = amountET.getText( ).toString( );

   ……..

   try {
     float amount = Float.parseFloat( amountString );
     mortgage.setAmount( amount );

     ……..
      } catch( NumberFormatException nfe ) {
     mortgage.setAmount( 100000.0f );
     mortgage.setRate( .035f );
   }
 }

  public void goBack( View v ) {
   _____
  _____

    }
}
```

Grading:

1. Submit the project in zip file (extension zip)
2. A pdf document that contains the content of the following files: AndroidMainifest.xml, activity_main.xml, activity_data.xml, themes.xml, MainActivity class, and DataActivity class
3. A zoom link or YouTube that demonstrate the user of the Mortgage Calculator app (less than 3 minutes). Write the link on the comment section in the Dropbox.
4. Write down the full names of each team members on the comment box.
5. Write down the contribution of each team members. All the team member must agree on the contribution of each team member.

Note:
App is running correctly and met all the requirements is 40 points, but

Incomplete item 1(-40 points)
Incomplete item 2 (-10 points)
Incomplete item 3 (-10 points)