1. Open Eclipse, using the location of your ADT folder as your workspace. You may also create another file on your C drive’s directory before opening Eclipse where you would need to browse your computer in order to use it as the workspace.

2. In the upper left hand corner click the “New” icon. Another display will pop up and you should click the “Android” folder, and then select “Android Application Project.” Click next.

3. Inside in the Project name enter in “MortgageCalculator.” Leave the other settings in their current state.

4. Click next past the application configuration and the image icon settings.
5. When you arrive at the “Create Activity” page make sure “Blank Activity” is checked and click finish.

6. At this point you have created your initial application folder and are ready to begin editing the app. The first thing that needs to done is editing the current main activity file. Go to the file directory located on the left side and click the folder titled “res.” Inside the res, or resource folder, you will see a folder named “layout.” Click on the activity_main.xml file and open it.

7. Now we need to change the layout from RelativeLayout to TableLayout. Change RelativeLayout to TableLayout in the layout header and footer tags.
8. Now we will begin to add components to the activity view. In the “Outline” panel, located on the right side of the IDE, you will see the TableLayout we implemented in the previous step. Right-click and select “Add Row.” Repeat this 5 times. After creating the rows we need to go and edit the IDs for consistency purposes. Right-click “tableRow1” and click “Edit ID.” Change the name to “tableRow0.” Repeat for the consecutive numbers. Your screen should look like this when you done:
9. We will now add the text and edit text fields. In the Form Widget panel under the Palette panel on the left side of the IDE, place a TextView in the slot for the “tableRow0” (to make sure you entered in the right slot pay attention to the Outline panel and the desired table row will be highlighted). Look in the Form Widget panel for the EditText component and place it on the tableRow0 as well. Repeat this for rows 1 through 3. (If you want you can add an extra row and in row 3 place an empty TextView. This will give a space between the “Periods” and “Payment Amt” fields)
10. After adding all the components go through and change the IDs and text values for consistency purposes. When you are done your screen should look like this. (Insert screenshot)

11. At this point we have finished the GUI for the app. We will now add functionality to it. In the file directory panel on the far left of the IDE open the “src” folder and open the “MainActivity.java” file.

12. Add the components we placed inside the GUI as instance variables of their perspective variable type.
13. Now within the `onCreate` method it is beneficial to check to see if app just started or is being restored from memory. To do that, check if the `savedInstanceState` is not null.

14. Now we need to reference the instance variables we initialized in step 12. To do this we need to use the `findViewById` method and parse the result with its appropriate variable type. Your `MainActivity` class should look like this:

15. It is now time to create listeners for the buttons. In the `MainActivity.java` class create a new instance of the `OnClickListener` class and call it `oclBtnCalc`. Next you would need to override the `onClick` method. Since you are doing the calculations, you need to get the values the user entered in and turn them into digits that could be used to calculate the payment by parsing. So create
double variables and call them principle and rate and create an int variable called periods. Next, set principle equal to:

```
Double.parseDouble(principleEditText.getText().toString());
```

Essentially you would do the same thing for the rate variable, changing the principleEditText to the rate variable's respective EditText property and dividing the result by 1200.

Set periods equal to:

```
Integer.parseInt(periodsEditText.getText().toString());
```

You should now create a double variable called payment to hold the calculated payment. Set it equal to:

```
(rate * principle) / (1 - Math.pow(1 + rate, -periods));
```

Once you have that you should edit the paymentEditText to reflect the calculate payment.

```java
//Create click listener for calculate button
OnClickListener oclBtnCalc = new OnClickListener()
{
    @Override
    public void onClick(View v) {
        double principle = Double.parseDouble(principleEditText.getText().toString());
        double rate = Double.parseDouble(rateEditText.getText().toString()) / 1200;
        int periods = Integer.parseInt(periodsEditText.getText().toString());

        double payment = (rate * principle) / (1 - Math.pow(1 + rate, -periods));
        paymentEditText.setText(String.format("%.2f", payment));
    }
};
```

16. It is now time to create the listener for the reset button. Starting out create another instance of OnClickListener class and call it oclBtnReset. Next, override the onClick method where you set each of the EditText properties back to zero. Your MainActivity should include this:

```java
OnClickListener oclBtnReset = new OnClickListener()
{
    @Override
    public void onClick(View v) {
        principleEditText.setText("0.00");
        rateEditText.setText("0.00");
        periodsEditText.setText("0");
        paymentEditText.setText("0.00");
    }
};
```
17. After that, all that is left is setting the OnClickListener for each of the buttons. To do that, include the following code after the reset button listener. 

```
calculateButton.setOnClickListener(oclBtnCalc);
resetButton.setOnClickListener(oclBtnReset);
```

18. We have now completed the code and are ready to run it. In the top tool bar click the green play button and run as an android application.