

# Java for Mobile Development

 Search this site

## Navigation

[Home](#)[Syllabus - Fall 2015](#)[Assignments - Fall 2015](#)[Content - Mini 1](#)[Content - Mini 2](#)[Assignments](#) >

## Assignment 2

Prepare an Android App using the interface shown at <http://www.mlcalc.com/>. Only code the Mortgage App and do not code the loan app.

1. •For calculations show:
  1. –Total Monthly Payment
  2. –Total of payment for Mortgage term
  3. –Payoff date
2. I realize the web calculator provides other options. You do not have include those in this submission (just the three requested in Point 1).
3. If you need help with formulas please visit - <http://learn-java-by-example.com/2010/java/simple-mortgage-calculator>
4. Help link on page <http://www.mlcalc.com/> provides access to basic terms for this app.
5. Submit your assignment to [cislabs04@gmail.com](mailto:cislabs04@gmail.com)

### Technical Requirements:

- You are required to show the calculations in one or two activities (No Fragment implementation yet)
- Populate the database with data generated for each calculation.
- Setup a separate tier for User Interface, Model and DB Access. These can be organized as follows:
  - UI package can have code for dynamically generated controls, coded intents for controlling activities and any other user interface related code.
  - Model package will define objects (for e.g,

Mortgage or MortgageCalc) that will act as intermediary between UI and DB Layers.

- Util package will have classes for object persistence and reading data to DB.
- So in other words the three layers will interact like this - For saving data you will create an Object that will get its properties populated from the UI and will be passed to a save function. The save function will read values from the object and map it to the columns in the DB.

### Grading Rubric

#	Criteria	Total Points	Points Awarded
		10	XX
1.	Program Specifications / Correctness	5	XX
1.	No errors, program always works correctly and meets the specification(s).		
2.	The code could be reused as a whole or each routine could be reused.		
3.	UI is user-friendly		
4.	Separate layers have been setup for Model, UI and Util as described in requirements		
2.	Readability	1	XX
1.	No errors, code is clean, understandable, and well-organized.		
2.	Code has been packaged and authored based on Java Coding Standards.		
3.	Documentation	1	XX
1.	The documentation is well written and clearly explains what the code is accomplishing and how.		
4.	Code Efficiency	3	XX
1.	No errors, code uses the best approach in every case.		
	The code is extremely efficient without sacrificing readability and understanding.		
2.	Ability to correct exceptions is added in exception handling class. Ability to log exceptions is added.		

### Comments