

## CS 501 Mobile Application Development

### Worksheet 2 – Events, Simple Programming Tasks

Due next class (9/20 or 9/22)

Date: 9/20/2022

Team Members: Alex Wang, Jinpeng Lyu, Lesley Chen, Tiffany Chen

The purpose of this worksheet is to begin considering what will be expected, in terms of scope and level of effort required for final projects. You will also practice what we have learned regarding Activities and Events.

This worksheet is to be done in collaboration with your project team. Everyone must participate, do not coast. Be sure you understand every part of the submission. Be sure to clearly list the people in your group when submitting. **Do not divide and conquer, you must meet and work together to solve the problems. I will be asking questions and expect everyone to be able to answer. If you cannot do this, please work individually.**

#### Submission Instructions:

Assignment will be submitted via Blackboard. NEVER email homework.

1. Ensure everyone's full name is neatly written at the top of the assignment.
2. For programming tasks, simply print out the java code, the xml layout, and include screenshots that show your app working, eg, before and after running. Include as many screenshots needed to make clear that your app works.
3. For non-programming tasks, eg, short answers, just type these up neatly.
4. Put your assignments in GitHub and submit the link.

Remember only one submission per group. Be sure to choose someone reliable to submit.

#### Part I:

##### 1. What are a few differences between the ART and the DVM?

ART uses Ahead of Time approach, takes more time to boot, has more storage, increases battery performance, has better garbage collection and better user experience while DVM uses Just In Time Approach, has more battery and CPU utilization, less storage, and longer app loading time.

##### 2. What is the difference between Early and Late Event Binding?

Early binding is a compile time process while Late Binding is a run-time process. In Early Binding, class information is used while in Late Binding, the object is used to resolve method calling. An example of Early Binding is method overloading, and an example of Late Binding is method overriding.

##### 3. What are some good routines/things to do (generally speaking) to include in the Activity's onCreate Event?

Some routine things to do include setting the layout with setContentView and initializing the variables declared in MainActivity.

#### 4. Why would a developer prefer to use Early vs. Late Binding of Events?

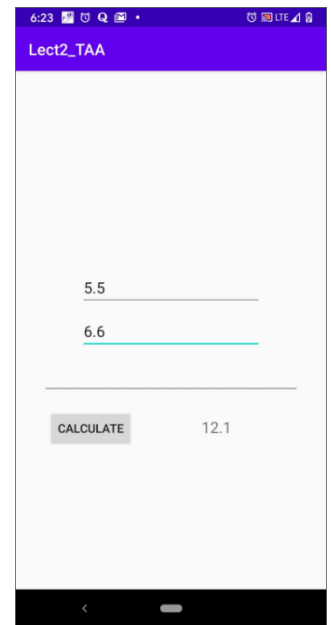
A developer would prefer to use Early vs Late Binding of events depending on what they want the program to do as Early binding increases program efficiency and speed, while Late Binding reduces program size, so a developer would make the choice depending on the advantages Early vs Late Binding offer.

### Part II: Simple Programing Task – Modify In Class Example

Modify the simple arithmetic program that was hard coded in lecture to add two operands. It must perform other operations, including Addition, Subtraction, Multiplication, Division, and Modulus.

Use any views of your choice to allow the user to select the operation. For example, you might use radio buttons to choose from a list of options, or a spinner view might be better. You decide how it should work, but make it usable and intuitive. As stated multiple times, this is a seminar class and will require bits of research to use things like radio buttons and/or spinners.

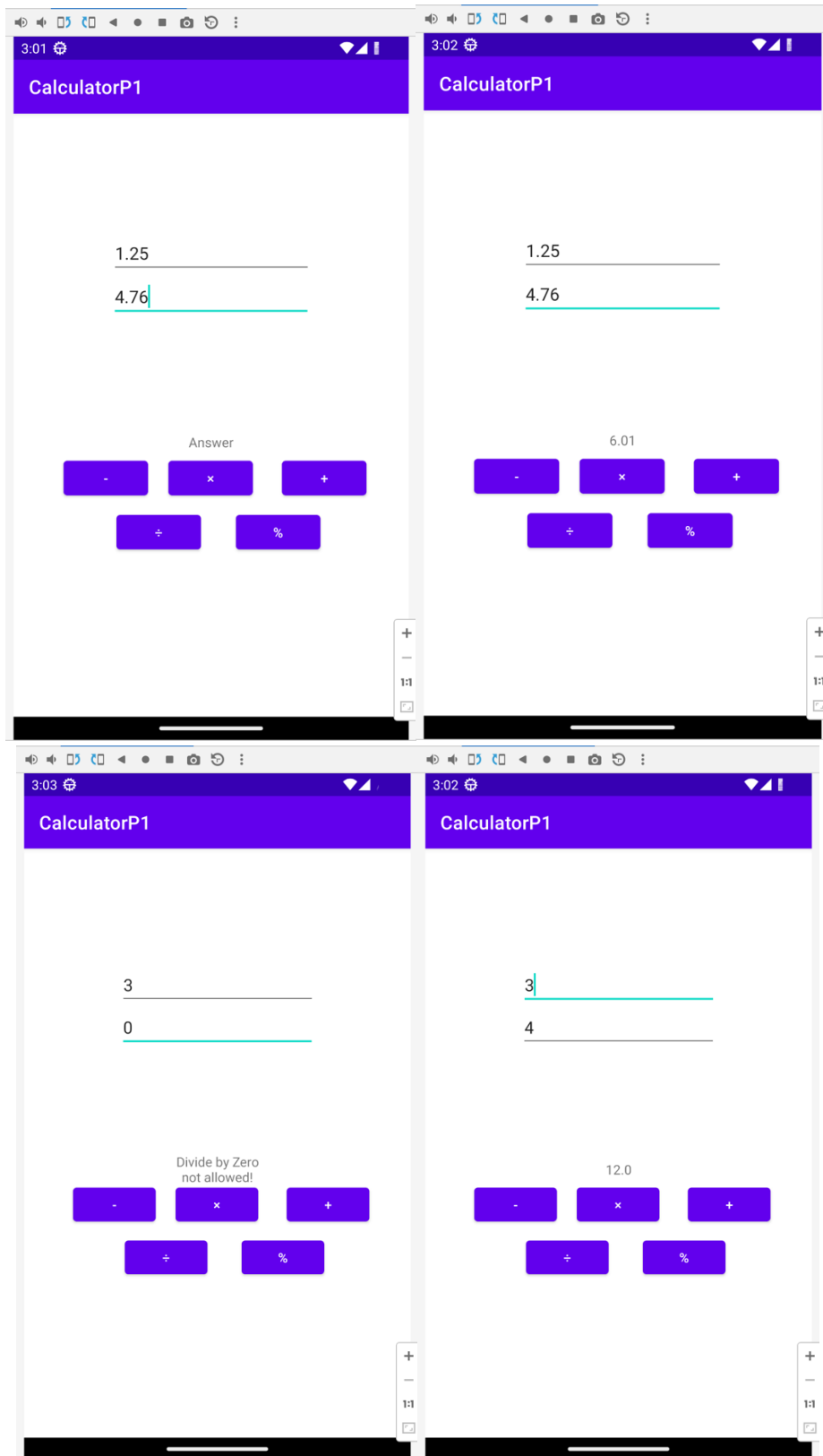
You must also handle error conditions, including divide by zero. A simple "invalid Operand" message will not be sufficient. Be creative, for example divide by zero can say something like, "Divide by Zero not allowed.", etc.



#### GitHub Link:

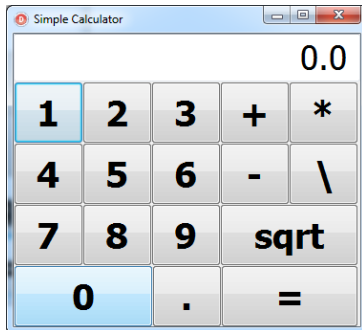
.java and .xml files: [https://github.com/AlxWang9966/WorkSheet\\_2/tree/master/Wksht2\\_part2\\_hwfiles](https://github.com/AlxWang9966/WorkSheet_2/tree/master/Wksht2_part2_hwfiles)  
all files : <https://github.com/AIAuB/CalculatorP1>

# Screenshot:



### Part III: Write A New App

Implement a calculator App for Android. The GUI should look very similar to the one shown below, you must use the ConstraintLayout as explained in class. It of course should work like a real calculator. Clicking on the numbers should update EditText at the top. Users should also be able to type numbers directly in the EditText. Be sure to handle all exceptions, including Divide by zero.



GitHub Link:

.java and .xml files:

[https://github.com/AlxWang9966/WorkSheet\\_2/tree/master/Wksht2\\_part3\\_hwfiles](https://github.com/AlxWang9966/WorkSheet_2/tree/master/Wksht2_part3_hwfiles)

all files: [https://github.com/AlxWang9966/WorkSheet\\_2](https://github.com/AlxWang9966/WorkSheet_2)

Screenshots:

