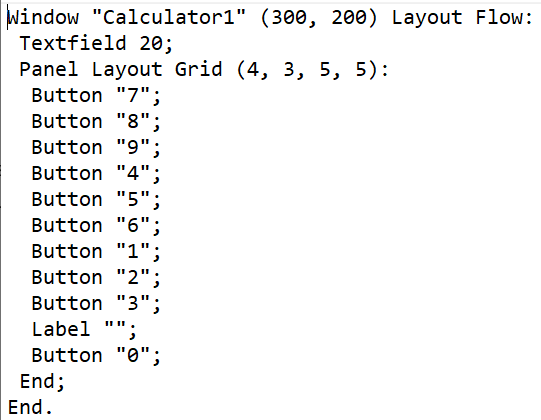
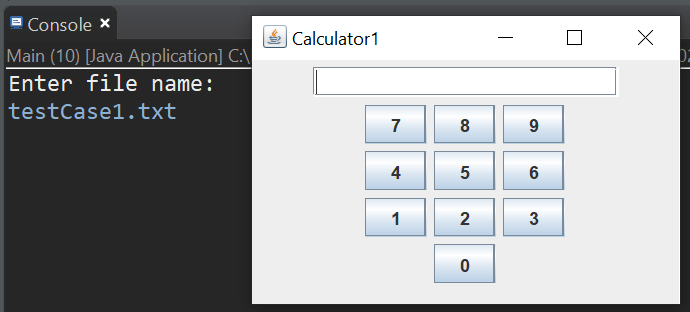
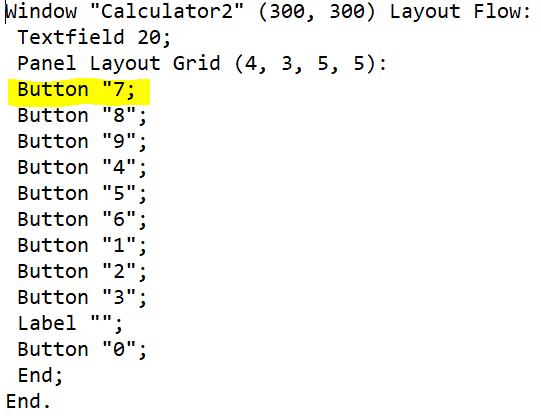
Kresimir Tokic  
CMSC330 Project 1  
Test Plan, Lessons Learned

**Test Plan**

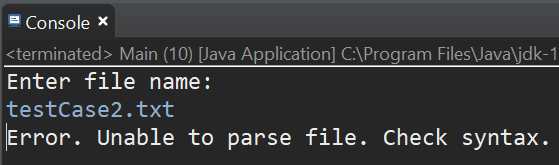
* This test case illustrates the successful parsing of a grammatically and syntactically correct file with two different layouts, buttons, text field and label.
* Test Case 1 sample input



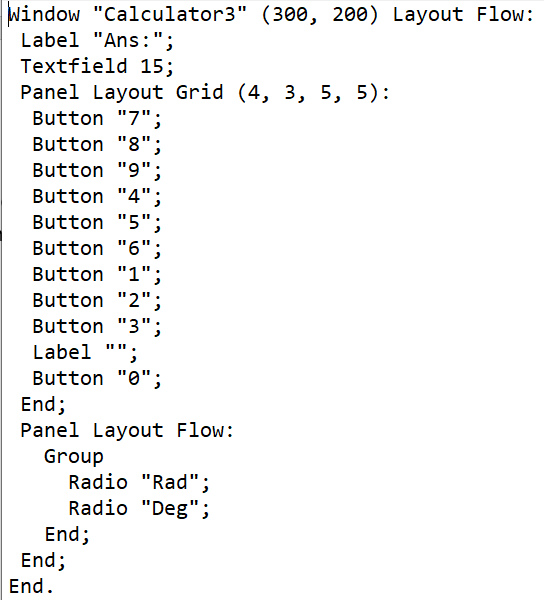
* Program output 
* This test case uses the same text file but has a syntax error at its innermost level.
* Test Case 2 sample input

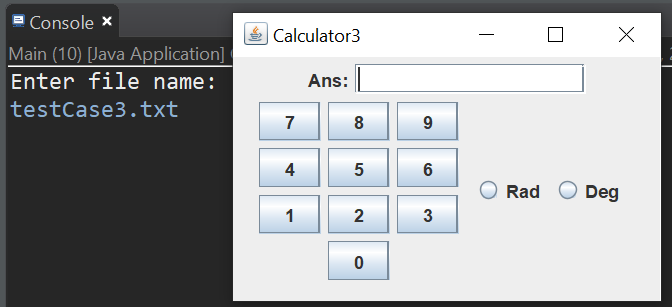


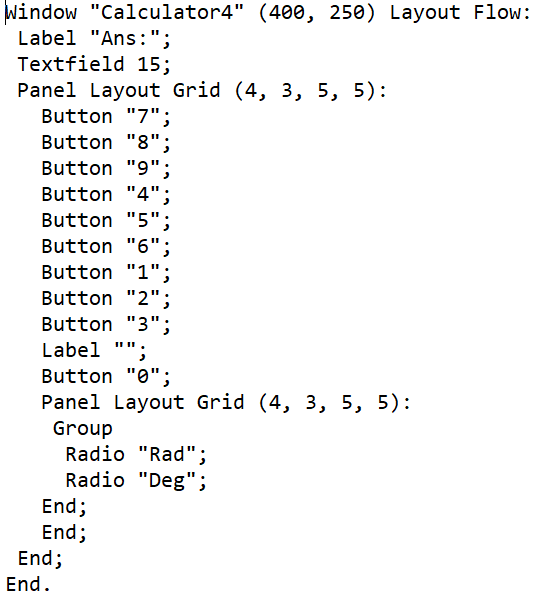
* Program Output



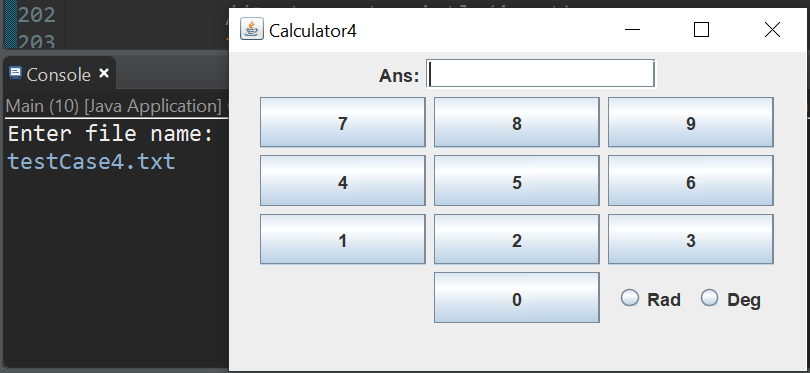
* This test case uses the same text file as test case 1 but with modified window dimensions, an additional label in the frame, a second non-nested panel including a group of radio buttons.
* Test Case 3 sample input



* Program Output:  
  
* This test case uses the same text file as test case 4 but instead nests the second panel.
* Test Case 4 sample input



* Program Output

  
  
I’m unsure why by button sizes changed with this test case. I was never able to figure that out.

**Lessons Learned**

I started this project quite confidently. I felt I had a good grasp on the algorithm necessary to build a recursive descent parser. I very quickly wrote my code using the sample test case to make sure I was on the right path. Everything was going smoothly until I re-read the requirements for nested panels and revaluated the grammar rules. I thought I had nailed it with my first attempt, but I really needed to draw out the grammar in BNF style to better visualize my logic and algorithm. I still haven’t learned how to sketch and pseudo-code my projects before diving headfirst into them. After finding the issues in my program, I refused to start scrap my code and start from scratch, so I reworked my code to accommodate the nested panels. I’ve never been a fan of Java’s layout managers; I generally prefer to hardcode the coordinates of my buttons but that’s not always the best solution either. I just find that’s a lot easier to prediction the output. I put a lot of print statements into my program to make sure it behaves the way I expect it to. I found a weird issue with Java. The statement System.out.println(frame.getLayout()); will print BorderLayout even if you’re using the flow layout. <https://stackoverflow.com/questions/34085129/jframesetlayoutlayoutmanager-not-working-forced-to-do-getcontentpane-setla>. I left my print statements commented out in my code just to illustrate this. Normally, I would prefer to clean my code and refactor as much as I think is necessary, but I decided to leave things as is to show my process and hope for constructive feedback. I found this project to be fun and challenging. I’m curious how similar it really is to write a compiler and how often a language will have grammatical ambiguities that could cause either compilation errors or other side effects. I chose to ignore case as much as possible but that was not explicitly stated as a requirement but someone else may chose to check case for syntactical correctness.