

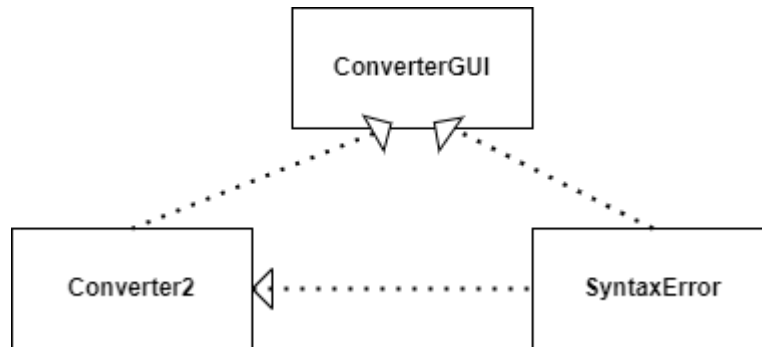
# Project 1 Test Report

Chloe Ingram

27 March 2022

CMSC 350 – 7380

## Part 1: UML Diagram



This diagram is meant to indicate that SyntaxError class is used by both ConverterGUI and Converter2 classes, and that Converter2 class is used by ConverterGUI class.

## Part 2: Test Cases

“Expected Output” values were generated using this online converter: <https://www.free-online-calculator-use.com/prefix-to-postfix-converter.html>

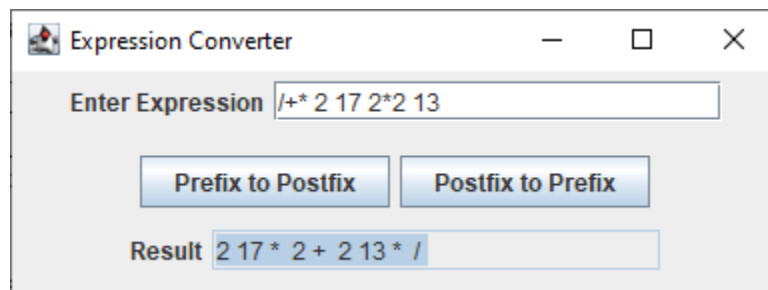


Figure 1. Test Case 1d

Prefix to Postfix Conversion					
Test Case No.	Infix Expression	Inputs	Expected Output	Actual Output	Pass/Fail
1a	$(3 * 7 + 11) / 12$	/ + * 3 7 11 12	3 7 * 11 + 12 /	3 7 * 11 + 12 /	Pass
1b	$(22 - 17 * 27) / 12$	/ - 22 * 17 27 12	22 17 27 * - 12 /	22 17 27 * - 12 /	Pass
1c*	$3*3+3/3$	+*3 3/3 3	3 3 * 3 3 / +	3 3 * 3 3 / +	Pass
1d*	$(2*17+2)/(2*13)$	/+* 2 17 2*2 13	2 17 * 2 + 2 13 * /	2 17 * 2 + 2 13 * /	Pass
*These test cases utilized no spaces with the exception of spaces between operands.					

“Expected Output” values were generated using this online converter: <https://www.free-online-calculator-use.com/postfix-to-prefix-converter.html>

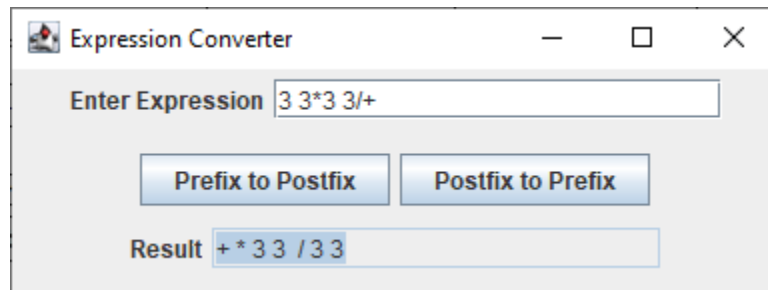


Figure 2. Test Case 2d

Postfix to Prefix Conversion					
Test Case No.	Infix Expression	Inputs	Expected Output	Actual Output	Pass/Fail
2a	$(27 + 13) / 2$	27 13 + 2 /	/ + 27 13 2	/ + 27 13 2	Pass
2b	$2 + 3 * 12 - 2 + 1$	2 3 12 * + 2 - 1 +	+ - + 2 * 3 12 2 1	+ - + 2 * 3 12 2 1	Pass
2c*	$2 + (8 + 3 * 24) / 2$	2 8 3 24*+2/+	+ 2 / + 8 * 3 24 2	+ 2 / + 8 * 3 24 2	Pass
2d*	$3*3+3/3$	3 3*3 3/+	+ * 3 3 / 3 3	+ * 3 3 / 3 3	Pass
*These test cases utilized no spaces apart from spaces between operands.					

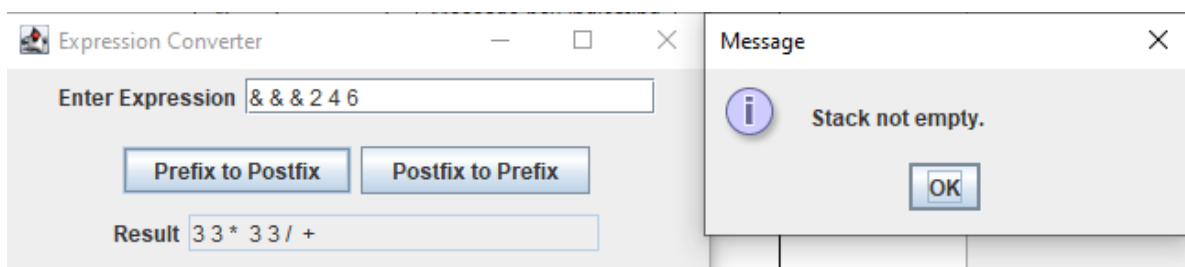


Figure 3. Test Case 3f

SyntaxError Checks					
Test Case No.	Testing Aspect	Inputs	Expected Output	Actual Output	Pass/Fail
3a	Invalid format for prefix to postfix	2 - 3 + 3 + + <i>*Prefix to Postfix button click*</i>	"Empty operand stack"	Message box indicating "Empty operand stack"	Pass
3b	Invalid format for postfix to prefix	- 2 * 11 <i>*Postfix to Prefix button click*</i>	"Empty operand stack"	Message box indicating "Empty operand stack"	Pass
3c	Invalid operator	2 3 & <i>*Postfix to Prefix button click*</i>	"Invalid operator"	Message box indicating "Invalid operator"	Pass
3d	Empty operand stack due to & symbol not being considered a delimiter and thus trying to parse expression "&234 +"	& 2 3 4 + <i>*Prefix to Postfix button click*</i>	"Empty operand stack"	Message box indicating "empty operand stack"	Pass
3e	Stack not empty due to invalid operator	& & & 2 4 6 <i>*Prefix to Postfix button click*</i>	"Stack not empty"	Message box indicating "stack not empty"	Pass
3f	Ensuring valid operation after error	& & & 2 4 6 <i>*Prefix to Postfix button click*</i> + * 3 3 / 3 3 <i>*Prefix to Postfix button click*</i>	"Stack not empty"  3 3 * 3 3 / +	Message box indicating "stack not empty"  3 3 * 3 3 / +	Pass
3g	Stack not empty due to too many operands	3 4 + 2 1 <i>*Prefix to Postfix button click*</i>	"Stack not empty"	Message box indicating "stack not empty"	Pass

GUI Functionality					
Test Case No.	Testing Aspect	Inputs	Expected Output	Actual Output	Pass/Fail
4a	Null expression value with button press	<i>*Prefix to Postfix button click*</i>	"Empty operand stack"	Message box indicating "empty operand stack"	Pass
4b	Null expression value with button press	<i>*Postfix to Prefix button click*</i>	"Empty operand stack"	Message box indicating "empty operand stack"	Pass
4c	ON_EXIT_CLOSE()	<i>*X press in top left corner*</i>	Code termination	<terminated>	Pass

### Part 3: Lessons Learned

I've never used a StringTokenizer before, and I think this will be incredibly useful in all future programs. Additionally, I learned the power of StackOverflow and the innate "okayness" with researching code. I spoke with some of my fellow engineers at work who said that's their primary workflow when trying to figure out a problem – research the problem, find some code, change that code to suit your needs, document. In this case, most of my research stemmed around my lack of understanding of the Stack class and some of the errors I was getting. Simple things like conversions from string-to-char to make my operator checking function work, "token.chars().allMatch( Character::isDigit )", and String var = var1 + ' ', were all things that I learned spending a huge amount of time in source documentation, and I think I should just download source documentation to make it easier to search for things. Also, I learned that some people create/use plugins in Eclipse to run scripts for debugging, as I really wanted to find a script that could check my code for code duplication. Unfortunately, I think my version of Eclipse is too new for most of the tutorials, so I couldn't get it to work.