

Lab Report #1

ECE 322

Yenosibina Makanjuola

1629453

Introduction (3 marks)

The purpose of this lab was to give a practical introduction to black box testing techniques. This lab focuses on dirty testing, error guessing, and partition testing. The completion of this lab exercise ensures that a person understands and is exposed to the various black box testing methods mentioned above.

Part One – Calculator Program (32 marks)

Q1: Require an explanation of the application under test, they should describe the problem that is being tested :

Answer: The application under testing is a calculator that performs simple arithmetic operations such as addition, subtraction, division, powers, and multiplication. The calculator has a GUI which contains values from 0-9, the bracket operators, the power operator, the division operator, multiplication operator, the addition operator, the subtraction operator, a clear button, delete button, and a dot to help represent decimal numbers. In this part we are tasked with finding errors in the calculator program by using Error Guessing and Failure/Dirty Testing.

Q2: Explain what testing methods you are using. The testing methods should be explained, what do they test, what are they good for etc ..

Errors in the application should be identified, and some justification given for what may be causing these errors. :

Answer:

1. The testing methods being used are failure/dirty testing and error guessing.
 - a. Failure/Dirty Testing: This is a form of testing that test all possible inputs and actions in the application to break the software. This form of testing basically tests for anything that could break the program and considers every input. It is good for testing whether the application can withstand invalid or unpredictable inputs.
 - b. Error guessing: This is a form testing that relies on experience and guesses for the location of errors based on common programming mistakes. This is used for testing common programming errors. It is good for finding errors in which formal testing methods could miss.
2. Justification of Errors:

- a. Test cases 25–28 failed because the program was unable to correctly prioritize the exponentiation operator (^) as having a higher precedence over other arithmetic operators. The standard order of operation used is BEDMAS
- b. Test 7 failed because the program doesn't implicitly infer that a multiplication sign should be used when no operator is put before a bracket
- c. Test 13 failed because the program assumes that an output must always be given if the equals button is clicked
- d. Test 16 and 18 failed because the delete button does not handle input from the keyboard properly
- e. Test 29-30 failed because the calculator automatically adds zero as the second operand if an operand was not given at the beginning.
- f. Test 32 failed because the calculator doesn't consider infinity as a valid output
- g. Test 41 and 42 failed because it surpasses the highest possible value that the calculator can compute

Q3: Discuss the effectiveness of the testing methods used

Answer:

1. Failure/Dirty Testing: This method is very effective when trying to find errors because it looks at every possible input that the program could have but it is very time-consuming testing for every single possible input. This works very well when combined with other testing techniques.
2. Error Guessing: The effectiveness on this method depends on the tester using it because it relies on intuition and experience. Therefore, the effectiveness of this testing method is proportional to how much experience the programmer has. This testing method usually looks for common user or programming errors such as invalid input and boundary analysis, this method should be combined with other testing methods because it is not very comprehensive.

Q4: Test Cases Table:

In the below test cases when I say enter numbers, it means click the number. It was noticed during testing that there's a difference in the application when the buttons are clicked or entered straight from the keyboard.

Answer:

ID	Inputs	Description	Expected Result	Actual Result
1	3*6	Test the multiplication operator <ol style="list-style-type: none"> 1. Enter two numbers but in-between those two numbers should be the multiplication operator 2. Click the equals button 	18.0	18.0

2	10-15	Test the subtraction operator 1. Enter two numbers but in-between those two numbers should be the subtraction operator 2. Click the equals button	-5.0	-5.0
3	120+1000	Test the addition operator 1. Enter two numbers but in-between those two numbers should be the addition operator 2. Click the equals button	1120.0	1120.0
4	21/3	Test the division operator 1. Enter two numbers but in-between those two numbers should be the division operator 2. Click the equals button	7.0	7.0
5	2^3	Test the power operator 1. Enter two numbers but in-between those two numbers should be the power operator 2. Click the equals button	8.0	8.0
6	2.3 * 5.6	Test the multiplication operator with decimals 1. Enter two decimal numbers but in-between those two numbers should be the multiplication operator 2. Click the equals button	12.88	12.88
7	2(3)	Test the bracket operator 1. Enter a number follow it with '(' 2. Then another number then close it with '3' 3. Click the equals button	6	23.0
8	+	Test the situation where no number is given for the addition sign. 1. Enter the addition operator	Nan	Nan

		2. Click the equals button		
9	-	Test the situation where no number is given for the subtraction sign. 1. Enter the subtraction operator 2. Click the equals button	Nan	Nan
10	*	Test the situation where no number is given for the multiplication sign 1. Enter the multiplication operator 2. Click the equals button	Nan	Nan
11	/	Test the situation where no number is given for the division sign. 1. Enter the division operator. 2. Click the equals button	Nan	Nan
12	()	Test the situation where no number is given for the bracket operator. 1. Enter the bracket operator "()". 2. Click the equal's button 3. Enter "(" 4. Click the equal's button 5. Enter ")" 6. Click the equal's button	Nan	Nan
13	No input given	Test the situation where no input is given and the input box is empty 1. Click the equals button	Blank result- The input box remains blank	0.0
14	555 * ffff	Test the situation where operations are performed using a number and letter 1. Enter a number 2. Enter the multiplication operator 3. Enter letters	Nan	Nan

		4. Click the equals button		
15	2 * 3	Test the delete button with the input being the button being clicked in the application <ol style="list-style-type: none"> 1. Click the respective operators and operands in the calculator 2. Press the delete button 3 time 	Empty result box	Empty result box
16	FF22	Test the delete button with input coming from the keyboard. <ol style="list-style-type: none"> 1. Enter the input on the left using the keyboard 2. Press the delete button 4 times 	Empty result box	The numbers did not delete – FF22
17	FF22	Test the delete button with input from the keyboard then input from clicking the button. <ol style="list-style-type: none"> 1. Enter FF using the keyboard 2. Enter 22 by clicking the button 3. Press the delete button 4 times 	Empty result box	Empty result box
18	22FF	Test the delete button when entering input by clicking the buttons then using the keyboard <ol style="list-style-type: none"> 1. Enter 22 by clicking buttons 2. Enter FF by using the keyboard 3. Press delete button twice 	22	2
19	(2 + 9) *4	Test Addition and Multiplication with brackets <ol style="list-style-type: none"> 1. Enter the input button by clicking the buttons 2. Click the equals button. 	44.0	44.0

20	$2 + 9 * 4$	Test Addition and Multiplication without brackets 1. Enter the input button by clicking the buttons 2. Click the equals button.	38.0	38.0
21	$11 - 13 * 4$	Test Subtraction and Multiplication without brackets 1. Enter the input button by clicking the buttons 2. Click the equals button.	-41.0	-41.0
22	$(13 - 11) * 4$	Test Multiplication and Subtraction without brackets 1. Enter the input button by clicking the buttons 2. Click the equals button.	8.0	8.0
23	$8 + 2 / 4 - 6$	Test Addition, Division and Subtraction without brackets 1. Enter the input button by clicking the buttons 2. Click the equals button	2.5	2.5
24	$8 * 6 / 2 - 11 + 6$	Test Addition, Multiplication, Division and Subtraction without brackets 1. Enter the input button by clicking the buttons 2. Click the equals button	19.0	19.0
25	$2 ^ 3 * 4$	Test Exponent and Multiplication without brackets 1. Enter the input button by clicking the buttons	32.0	4096.0
26	$2 ^ 3 / 4$	Test Exponent and Division without brackets 1. Enter the input button by clicking the buttons	2.0	1.6818

27	$2^3 + 4$	Test Exponent and Addition without brackets 1. Enter the input button by clicking the buttons	12.0	128.0
28	$2^3 - 4$	Test Exponent and Subtraction without brackets 1. Enter the input button by clicking the buttons	4	0.5
29	*6	Test multiplication operator before number 1. Enter input by clicking buttons 2. Click the equals button	Invalid input or Nan	0.0
30	/6	Test Division operator before number 1. Enter input by clicking buttons 2. Click the equals button	Invalid input or Nan	0.0
31	()6	Test bracket operator before number 1. Enter input by clicking buttons 2. Click the equals button	Nan	Nan
32	1/0	Test Division by zero 1. Enter input by clicking buttons 2. Click the equals button	Inf	Nan
33	+6	Test addition operator before number 1. Enter input by clicking buttons 2. Click the equals button	6.0	6.0
34	-6	Test subtraction operator before number 1. Enter input by clicking buttons 2. Click the equals button	-6.0	-6.0
35	a. Sdfd222sdfs b. 22222 c. Fffff22	Test the clear button from input sources coming from	Blank result	Blank result

		both the keyboard and by clicking the input <ol style="list-style-type: none"> 1. Enter the first input from the keyboard 2. Click Clear 3. Enter the second input by clicking 4. Click Clear 5. Enter the third input by clicking and from the keyboard. 6. Click Clear 		
36	3++3	Test two addition operands before next number <ol style="list-style-type: none"> 1. Enter input by clicking buttons 2. Click the equals button 	Invalid input or not a number	6.0
37	3//3	Test two division operands before next number <ol style="list-style-type: none"> 1. Enter input by clicking buttons 2. Click the equals button 	Invalid input or Nan	Nan
38	3**3	Test two multiplication operands before next number <ol style="list-style-type: none"> 1. Enter input by clicking buttons 2. Click the equals button 	Invalid input or not a number	0.0
39	3--3	Test two subtraction operands before next number <ol style="list-style-type: none"> 1. Enter input by clicking buttons 2. Click the equals button 	Invalid input or Nan	Nan
40	3^^3	Test two power operands before next number <ol style="list-style-type: none"> 1. Enter input by clicking buttons 2. Click the equals button 	Invalid input or Nan	1.0
41	(3.4)*(10^38) *2	Test the highest possible value that can be computed <ol style="list-style-type: none"> 1. Enter input by clicking buttons 2. Click the equals button 	4.8*10^38	Nan

42	$-(3.4) \times (10^{38}) \times 2$	Test the lowest possible value that can be computed 1. Enter input by clicking buttons 2. Click the equals button	-4.8×10^{38}	Nan
43	6×0	Test zero multiplication 1. Enter input by clicking buttons 2. Click the equals button	0.0	0.0
44	a. 6+ b. 6* c. 6- d. 6/ e. 6^	Test the case where the input operator is at the end 1. Enter input by clicking buttons 2. Click the equal's button 3. Repeat n times where n is the number of inputs	Nan	Nan
45	Delete button	Test the delete button when no input has been given 1. Click the delete button	Blank result	Blank result
46	Clear button	Test the clear button when no input has been given 1. Click the clear button	Blank result	Blank result

Part 2 – Triangle Classification Program (32 marks)

Q1: explanation of the application under test, and a description of the problem

Answer:

The application under test is an application that checks the length of the 3 sides of a triangle and determines what type of triangle it is if it is valid or determines whether it is possible to make a triangle with the given length. In this part we are tasked to use Partition Based Testing to find errors in the program.

Q2: List all the Partition classes

Answer:

Partition Classes:

Valid Equivalent classes

1. $a + b > c$

2. $a + c > b$
3. $b + c > a$
4. All inputs fall within a valid range on INT

Invalid Equivalent classes

1. Number of arguments > 3
2. Number of arguments < 3
3. Int out of range
4. Non integer values (decimals, characters)
5. $a + b \leq c$
6. $a + c \leq b$
7. $b + c \leq a$

Q3: Identification of error(s) in the program. Must be identified and justified

Answer:

The code was improperly written when testing for isosceles triangles it used $a + b \geq c$ instead of $a + b > c$. This is the assumed error in the code because tests 9, 10, 11 failed when the test cases involved summing the inputs to an exact value but for test cases where the values were greater – test 1,2,3 and where the values were less than test 14-16 the program behaved accordingly.

Q4: Discuss the effectiveness of the testing methods used

Answer: Partition based testing is very effective because it covers as many of test cases with minimal amounts of test, but it may tend to miss some edge cases. Therefore, this testing method should be combined with other testing methods

Q5: Test Cases Table :

Answer:

ID	Input a	Input b	Input c	Description	Expected Result	Actual Result
1	10	11	12	Test the situation where $a + b > c$	Scalene	Scalene
2	13	9	13	Test the situation where $a + c > b$	Isosceles	Isosceles
3	20	20	20	Test the situation	Equilateral	Equilateral

				where $b + c > a$		
4	100	200	150	Test the case where all inputs fall with a valid range less than INT_MAX	Scalene	Scalene
5	'a'	5	6	Test the situation where numbers are combined with characters	ERROR: Invalid argument - non integer	ERROR: Invalid argument - non integer
6	-2	6	8	Test the situation where negative numbers are used.	ERROR: Invalid argument - non positive value	ERROR: Invalid argument - non positive value
7	2147483648	6	2147483648	Test the situation where numbers greater than INT_MAX are used.	ERROR: Invalid argument - non integer	ERROR: Invalid argument - non integer
8	3.5	5	6	Test the situation where a decimal value is included in the input	ERROR: Invalid argument - non integer	ERROR: Invalid argument - non integer
9	1	1	2	Test Values where $a + b = c$	ERROR: Invalid triangle	Isosceles
10	3	5	2	Test Values where $a + c = b$	ERROR: Invalid triangle	Scalene

11	6	6	12	Test Values where $b + c = a$	ERROR: Invalid triangle	Isosceles
12	4	5		Test values where number of arguments is less than 3	ERROR: Not enough arguments	ERROR: Not enough arguments
13	4	5	6 7	Test values where number of arguments is greater than 3	ERROR: Too many arguments	ERROR: Too many arguments
14	10	4	4	Test case where $b + c < a$	ERROR: Invalid triangle	ERROR: Invalid triangle
15	2	7	3	Test the case where $a + c < b$	ERROR: Invalid triangle	ERROR: Invalid triangle
16	2	2	8	Test the case where $a + b < c$	ERROR: Invalid triangle	ERROR: Invalid triangle

Conclusion (3 marks)

In conclusion, the objectives of this lab were achieved by testing both the Calculator and Triangle programs using various black-box testing techniques. In Part 1, error guessing and failure testing revealed the main issue with the Calculator program: it failed to follow the correct order of operations (BEDMAS), resulting in incorrect calculations. In Part 2, partition-based testing of the Triangle program identified an error in the use of inequality signs, leading to incorrect triangle classification. These tests allowed us to uncover bugs and understand the reasons behind them.