ECE 325 OBJECT-ORIENTED SOFWARE DES (LEC A1 Fa18)

Assignment 3: Exception Handling

·Source code: *Calculator.java*

·Due date: **Thursday 4th of October (5:00 pm)**. A working copy of your solution must be submitted to eClass before this date.

Part 1: Calculator

Your program must be able to read in the following expressions. You may wish to construct a BNF grammar in the style of the cookie exercise as an initial task.

Compile, run, and test your program with at least these expressions:

```
let x = 1;
(let x = 1) + x;
(let a = 2) + 3 * a - 5;
(let x = (let y = (let z = 1))) + x + y + z;
1 + (let x = 1) + (let y = 2) + (1 + x) * (1 + y) - (let x = y) - (let y = 1) - x;
1 + (let a = (let b = 1) + b) + a + 1;
(let a = (let a = (let a = (let a = 2) + a) + a) + a) - 9;
(let x = 2) ^ (let y = 3);
(let y = 3) ^ (let x = 2);
```

Correct return values are 1, 2, 3, 4, 5, 6, 7, 8, and 9 respectively.

Part 2: Exception Handling

Add Java exception handling to your code by defining two exception classes `SyntaxError` and `RuntimeError`.
·A `SyntaxError` exception should be thrown when an illegal character is found, a closing `)` is not found, or `a =` is not used in a `let` expression.

·A `RuntimeError` exception should be thrown when an identifier is encountered for which no value can be found.

The exceptions should propagate the error to the main program which prints the diagnostics of the error. You must handle these errors using Java exceptions and the message should be printed by an exception handler in a catch clause.

These will be the exception test cases:

```
1 + (2 * 3;                     // syntax error: ')' expected
(let x 5) + x;                  // syntax error: '=' expected
(let x = 5) (let y = 6);        // syntax error: operator expected
(let x = 5 let y = 6);          // syntax error: ')' expected
(ler x = 5) ^ (let y = 6);      // runtime error: 'ler' undefined
(let x = 5) + y;                // runtime error: 'y' undefined
```

A Working Procedure Example

```
1 + (let x = 1) + (let y = 2) + (1 + x * (1 + y)) - (let x = y) - x;
```

	Expression	Stack	Pop & Return	Ha
1	1 +	| 1 +|	/	<E
2	1 + (let x = 1	| let x = 1| | 1 +|	/	x
3	1 + (let x = 1)	| 1 + 1|	1 (returned from "let x = 1")	x
4	1 + (let x = 1) + (let y = 2) + (1 + x * (1 + y	| 1 + y| | 1 + x *| | 1 + 1 + 2 + |	/ ("let y = 2" has already returned 2)	x y
5	1 + (let x = 1) + (let y = 2) + (1 + x * (1 + y)	| 1 + x * 3| | 1 + 1 + 2 +|	3 (returned from "1 + y")	x y
6	1 + (let x = 1) + (let y = 2) + (1 + x * (1 + y)) - (let x = y	| let x = y| | 1 + 1 + 2 + 4 - |	/ ("1 + x * 3" has already returned 4)	x y
7	1 + (let x = 1) + (let y = 2) + (1 + x * (1 + y)) - (let x = y) - x	| 1 + 1 + 2 + 4 - 2 - 2 |	/ ("let x = y" has already returned 2)	x y

8	`1 + (let x = 1) + (let y = 2) + (1 + x * (1 + y)) - (let x = y) - x;`	`<Empty>`	4	x y

Calculator.java

Submission status

Attempt number	This is attempt 1 (1 attempts allowed).
Submission status	Submitted for grading
Grading status	Graded
Due date	Thursday, 4 October 2018, 5:00 PM
Time remaining	Assignment was submitted 1 hour 9 mins early
Last modified	Thursday, 4 October 2018, 3:50 PM
File submissions	Calculator.java Export to portfolio
Submission comments	Comments (0)

Edit submission

Make changes to your submission

Feedback

Grade	30.00 / 30.00
Graded on	Tuesday, 9 October 2018, 12:23 PM

Graded by Aindrila Ghosh

You are logged in as **Arun Woosaree** (**Log out**)
ECE 325 (LEC A1 Fa18)

Help
Email