# CS2400 Spring 2022 Project 2

Total points: 100

Due date: Friday, March 18, 2022

## Purpose:

1. Master the interface and implementations of Stacks.
2. Use Stacks to solve some computer science problems.

“Please start working on this assignment as early as possible!”

## Task Descriptions:

Chapter 5 of the textbook (5th Edition) presents a couple of algebraic expression-related problems that could be solved by Stacks. In this assignment, you will **read and study** the textbook from page 164 to page 170 to learn

- the process of using stack in *Transforming an Infix Expression to a Postfix Expression*, and
- the process of using stack in *Evaluating Postfix Expressions*

and **implement** them in Java. Specifically,

(10 pts) **Task 1: Read and Study** the textbook from page 164 to page 168, “A Problem Solved: Transforming an Infix Expression to a Postfix Expression”. **Use** a stack to manually convert the following infix expression to postfix expression:

$$a * b \ / (c - a) + d * e$$

Note: **Your answer sheet must show** the conversion process step by step, presenting the status of the stack after each step using a table with three columns (like Figure 5-9 on page 168 of the textbook).

(20 pts) **Task 2:** Use a **LinkedStack** class instance to **implement** the algorithm ***convertToPostfix***, given in Segment 5.16 of the textbook. Note: the class **LinkedStack** presents an implementation of the ADT stack using a linked chain.

(20 pts) **Task 3:** Write a main method to **test** your implementation (Task 2) using the infix expression given in Task 1.

(10 pts) **Task 4: Read and Study** the textbook from page 168 to page 170, “A Problem Solved: Evaluating Postfix Expressions”. **Use** a stack to manually evaluate the postfix expression obtained from Task 1, assuming that $a = 2, b = 3, c = 4, d = 5$, and $e = 6$.

Note: **Your answer sheet must show** the evaluation process step by step, presenting the status of the stack after each step (like Figure 5-11 on page 169 of the textbook).

(20 pts) **Task 5:** Use a **ResizeableArrayStack** class instance to **implement** the algorithm ***evaluatePostfix***, given in Segment 5.18 of the textbook. Note: the class **ResizeableArrayStack** presents an implementation of the ADT stack using a resizable array.

(20 pts) **Task 6:** Write a main method to test your implementation (Task 5) using the postfix expression and the variable values given in Task 4.

## What to Submit?

1. Answer sheets for Task 1 and Task 4, which are "Task1.pdf" and "Task4.pdf" in pdf format.
2. Source codes for Tasks 2-3 and Tasks 5-6, which are *"StackInterface.java", "ResizeableArrayStack.java", "LinkedStack.java", "ArrayStackTest.java", and "LinkedStackTest.java".*

   Note:

   - The *class Node* could be used as a member inner class of the class *LinkedStack.* Alternatively, it is fine to make *class Node<T>* an independent class and place in an additional java file "Node.java".
   - Please properly comment java code to improve the readability.

   Please test your source codes using the **Eclipse** IDE and see if the codes are executable. Non-executable programs will result in a grade of zero.

3. Please zip all documents (if "Node.java" used, there are 8 files in total. 7 files otherwise.) as yourname_p2.zip and submit it on Canvas.

   Note: You will be graded based on the quality of your program.

## Textbook used:

- Data Structures and Abstractions with Java, 5th Edition, by Frank M. Carrano and Timothy M. Henry, Prentice Hall, 2018.