**COMP 200: Data Structures and Algorithms**

**Fall 2020**

**Lab 4: Stacks**

**Task 1:** Implement the ArrayStack class from page 233 of the textbook.

**Task 2:** Implement a function with signature transfer(S, T) that transfers all elements from stack S onto stack T, so that the element that starts at the top of S is the first to be inserted onto T, and the element at the bottom of S ends up at the top of T.

**Task 3:** Implement a function that reverses a list of elements by pushing them onto a stack in one order, and writing them back to the list in reversed order.

**Task 4:** Show how to use the transfer function, described in Task 2, and two temporary stacks, to replace the contents of a given stack S with those same elements, but in reversed order.

**Task 5:** Postfix notation is an unambiguous way of writing an arithmetic expression without parentheses. It is defined so that if “(exp1) **op** (exp2)” is a normal, fully parenthesized expression whose operation is **op**, the postfix version of this is “pexp1 pexp2 **op**”, where pexp1 is the postfix version of exp1 and pexp2 is the postfix version of exp2. The postfix version of a single number or variable is just that number or variable. For example, the postfix version of “((5 + 2) ∗ (8 − 3))/4” is “5 2 + 8 3 − ∗ 4 /”. Implement a non-recursive program that can input an expression in postfix notation and output its value.