Date: 21-Mar-2022

Core Java Programming Lab
Assignment-I

- Create two classes:
  - PfCalculator (Post-fix Calculator)
  - TestClac (contains main method)
- TestCalc should take input from the user in the following format examples(spaces in between tokens):
  - `1 2 + 3 4 + /`
  - `23 5 * 89 68 - 3 7 % / /`
- TestCalc should tokenize the given string and pass the tokens in a queue collection to PfCalculator's method.
- Handle all the appropriate exceptions.
- PfCalculator should make use of stack and queue collection implementations from java collection framework.
- Support these binary operators: plus(+), minus(-), multiply(*), divide(/), modulus(%) and exponentiation(^). In exponentiation, the sequence 10 2 ^ would result in 100 -- the 2nd operand is the exponent.
- All the operands can be doubles.
- The implementation of Post-Fix Calculator goes like this:
  - Let PfCalculator's evaluate method take Queue object as parameter.
  - Remove each element from the "input queue" one at a time and push it to Stack object. If the element is recognized as an operator, pop 2 string values off of the Stack, convert them to their numeric equivalents, perform the calculation with those values, and push the result (as a string) back onto the Stack. Otherwise, simply push the token onto the stack. When the input queue is empty, then pop the stack for the result and return it.

Note: Use at least these following concepts: packages, Exception Handling, Collections, Generics, Static, appropriate access modifiers, naming conventions.