Infix to Postfix Converter

Learning Goals

1. Develop your ability to use stacks, exceptions, and tokens in Java.
2. Develop your understanding of polymorphism.

The Algorithm

1. Create a BCAStack called **opstack** for keeping operators.

2. Create an empty list (or String) for **output**.

3. Convert the input to tokens.

4. While there are tokens remaining in the input:
   - If the token is an operand, append it to the end of the **output**.

   - If the token is a left parenthesis, push it on the **opstack**.

   - If the token is a right parenthesis, pop the **opstack** until the corresponding left parenthesis is removed. Append each operator to the **output**.

   - If the token is an operator, push it onto the **opstack**. However, first remove any operators already on the **opstack** that have higher or equal precedence and append them to the **output**.

5. When the input expression has been completely processed, check the **opstack**. Any operators still on the stack can be removed and appended to the end of the output list.

The algorithm on this page is from:
http://interactivepython.org/runestone/static/pythonds/BasicDS/InfixPrefixandPostfixExpressions.html