

CSIS0396A/COMP2396A - Assignment 1

Due: 17th Sep, 2013 23:30

Introduction

This assignment tests your basic programming skill in Java.

You are asked to write a **infix to postfix convertor**. Although program design will not be evaluated in this assignment, you are encouraged to apply the object-oriented programming technique that you have learnt so far.

You are also required to write **JavaDoc** for all non-private classes and non-private class members. **Programs without JavaDoc will not be marked.**

Task

You will be provided a skeleton file, `Infix2Postfix.java`, which consists of a simple `main()` method that **controls the basic program flow**; and a method that **reads a line from input**, which is implemented for you. You will need to implement the `doConversion()` method that **read an infix from input** (using the provided `readInfix()` method), convert that to **postfix** and **print** it out.

To support the **infix to postfix conversion**, you are required to implement the class **Stack**. You are **not** allowed to use any class provided in the packages in `java.util.*` or any other Java classes that provide the implementation of a stack from the Internet. Please refer to the tutorial for the basic operations of a stack.

Requirements

1. Your program should read in an arithmetic expression in **infix** form, and output the same expression in **postfix** form. Numbers and operators in the expression are separated by a **single space**.
2. Your program should support the **four arithmetic operators**, `+`, `-`, `*` and `/`. The **precedence** of operators `*` and `/` should be **higher than** that of operators `+` and `-`.
3. Your program should support the use of **parenthesis**. (Note the spacing before and after the parenthesis.)
4. Finally, your program should support **negative values**.

```
Input infix: 12 + 23
Postfix: 12 23 +
Input infix: 34 * 56
Postfix: 34 56 *
```

```
Input infix: 1 + 2 * 3
Postfix: 1 2 3 * +
```

```
Input infix: ( 1 + 2 ) * 3
Postfix: 1 2 + 3 *
```

```
Input infix: 1 + -2 * 3
Postfix: 1 -2 3 * +
```

Marking

- **60% marks** are given to the **functionality** of your program.
 - You may add **additional classes, instant variables and methods** to the class.
 - You may assume that the input is always a **valid, single-space** delimited **infix** expression.
 - Your program output must be **identical** to what is described in this document, with the exception of the trailing spaces at the end of each line of output.
- **40% marks** are given to your **JavaDoc**. A complete JavaDoc includes documentation of every classes, member fields and methods that are not private. JavaDoc for the main method may be omitted.

Submission:

Please submit *Infix2Postfix.java* to Moodle. **Late submission is not allowed.**

Do not submit .class file.

-- END --