CM 10227/50258: Getting Started With SRPN

Dr. Rachid Hourizi and Dr. Michael Wright

November 11, 2016

Introduction

- How can you make a start on the first large coursework
  - When you dont know much Java?
  - When you havent previously written a larger piece of code than that required by the lab sheets?

Read the Specification

Read the specification for SRPN, which includes but is not limited to:

- Your task is to write a program, which matches the functionality of SRPN as closely as possible.
- Note that this includes not adding or enhancing existing features.
- SRPN is a reverse polish notation calculator
- with the extra feature that all arithmetic is saturated
- i.e. when it reaches the maximum value that can be stored in a variable,
- it stays at the maximum rather than wrapping around.
- Our marking script will be the same but will use different test-cases.
- The program includes the less obvious features of srpn.

Understand RPN

- Reverse Polish notation (RPN) is a mathematical notation in which every operator follows all of its operands

$5 \quad 4 \quad + \quad =$

$3 \quad 2 \quad * \quad 4 \quad 4 \quad * \quad + \quad =$

Explore Our Code

```
10
2
+
=

3 3 * 4 4 * + =

etc
```

Code Iteratively

- Dont try to write a program that satisfies the whole specification immeditately
- Write a program that satisfies **some** of the specification, get it working, **save it** and then add functionality

Find a Small Enough SubProblem To Tackle In Version 1

Find a Small Enough SubProblem To Tackle In Version 1 e.g.

- Get any Java program working?
- Get a Program working that prints to the screen without taking user input?
- Get a program working that prints integers to the screen when they are entered neatly, one per line
- Get a program working that also recognises the difference between operators and integers
- Get a program working that also performs operations on integers before printing
- Think about an edge case
- Think abut another edge case
- etc

Review Our Discussion of Stacks

Review Our Discussion of Stacks

- You might find the concept of a Stack useful when writing SRPN
- We have seen stacks before
- Stacks are first in last out (FILO) data structures
- i.e. collections of data that give you back the last item that you put in

- ASIDE: It may be helpful to compare FILO with First In First Out (FIFO) Structures
- A Queue is a FIFO data structures

Review Our Discussion of Stacks

- Stacks provide the following functionality
  - Object push(Object element)
  - Pushes the element onto the stack.
  - boolean empty()
  - Tests whether stack is empty. Returns true if the stack is empty, and returns false if the stack contains elements.
  - Object peek( )
  - Returns the element on the top of the stack, but does not remove it.
  - Object pop( )
  - Returns the element on the top of the stack, removing it in the process.

Review Our Discussion of Stacks

- Java provides a prewritten Stack class
- i.e. a pre-written template from which you can create Stack Objects
- http://tinyurl.com/hewobm3f

- You may use snippets of other people's code
- but don't forget to provide a reference in comments

- We dont usually suggest using code that you dont fully understand
- You may make an exception in this case