

# CSE250 Fall 2016

## Assignment A5 – Desk Calculator

Due: 11/13/2016, 11:59PM

Last updated: 2016-11-02 22:40

### *Objectives*

- Practice design and implementation of stack-based algorithms.
- Implement a useful tool.

### *Introduction*

`dc` is one of the oldest UNIX commands. Despite its somewhat cumbersome syntax, it is still loved by many hard-core UNIX users. `dc` employs Reverse Polish Notation (RPN), supports arbitrary precision as well as macros and includes quite complex functions.

Your task in this assignment is to implement a `dc`-like tool. Specifically, you have to implement a function that for a sequence of tokens in RPN will check if the input expression is correct, and if yes it will evaluate it. You can assume that operands are of type `double` and only four basic binary operators `+` `-` `*` `/` are in use. You do not have to check correctness of tokens, however, you have to check if tokens form a valid RPN expression. To signal that an input expression is not valid you will have to use exceptions (see below).

To use exceptions, you have to include `<stdexcept>` header. To emit an exception that is related to “run-time” error the syntax is:

```
throw std::runtime_error("some useful message");
```

For example:

```
#include <stdexcept>

int main(int argc, char* argv[]) {
    if (argc == 1) throw std::runtime_error("argc is tooooo small");
    return 0;
}
```

### *Instructions*

1. Create directory A5 where you will place your code.
2. Download A5-handout.tar from:  
<http://www.jzola.org/courses/2016/Fall/CSE250/A/A5-handout.tar>.
3. Untar handout, and move `a5.cpp`, `a5.hpp` and `token.hpp` to your A5 directory. These files provide all functionality to: parse standard input into a sequence of tokens, handle exceptions and invoke your RPN code. Additionally, `token.hpp` provides interface to handle tokens (see below).

4. In `a5.hpp` implement function `RPN` considering the following:
- (a) `[first, last)` represents the range of input tokens.
  - (b) Each element in the range is of type `token` that is implemented in `token.hpp`.
  - (c) Class `token` provides the following methods:
    - `bool is_operator() const`; – returns true if token is an operator.
    - `bool is_operand() const`; – returns true if token is an operand.
    - `char as_operator() const`; – if the token is an operator returns character '+' or '-' or '\*' or '/' representing the operator, otherwise behavior is undefined.
    - `double as_operand() const`; – if the token is an operand returns value of the operand, otherwise behavior is undefined.
  - (d) If `[first, last)` is a valid RPN expression, return value should be the result of the evaluation of this expression.
  - (e) If `[first, last)` is not a valid RPN expression, or divide by zero occurs, `std::runtime_exception` should be thrown with any message.
  - (f) The only file you can edit is `a5.hpp`, and all your code must be contained in this file.
  - (g) To test your code you can pass RPN expression via the standard input to your `a5`. For example, these are equivalent invocations:
    - `./a5 <<< "2.71 3.14 + 2.68 *"`
    - `echo "2.71 3.14 + 2.68 *" | ./a5`that should produce 15.678. For an invalid RPN expression, or expression with division by zero, you should see one word only: error.

### Submission

1. Remove your binary code and other unrelated files (e.g. your test files).
2. Create a tarball with your `A5` folder.
3. Follow to <https://autograder.cse.buffalo.edu> and submit `A5.tar` for grading.
4. You have unlimited number of submissions, however, any submission after the deadline will have 50% points deducted.

### Grading

- 10pt: `a5.cpp` compiles with your `a5.hpp`, runs and has no memory leaks.
- 90pt: If you pass the initial test, there will be nine benchmark tests. You will get 10pt for each correctly completed test.
- If your code is **extremely** inefficient, for instance due to infinite loop, autograder will terminate your code and you will receive 0pt.

### Remarks

- Make sure that all file and directory names are exactly as instructed. Otherwise the grading system will miss your submission and you will get 0pt.
- Are you even reading this section?