

COMP322: Assignment 4 - Winter 2010

Due at 11:59pm EST, 14 Apr 2010

1 Introduction

In this assignment, we'll return to our simple calculator program. We will extend it to use C++ *exceptions* for error handling.

2 Requirements

Modify the `main()` program of your calculator to enclose the parsing code in a `try` block. Then add appropriate exception handlers inside one or more `catch` blocks.

You'll then add `throw` statements in all the appropriate places necessary to identify and report (at least) the following errors:

1. Division by zero
2. Undefined symbol in expression
3. General syntax error
4. Unmatched parenthesis

Your exceptions must be class types, but you are free to structure them in any reasonable fashion. You can simply create a single "Exception" class which contains information about the type of error, or you can create various distinct (but possible related) classes for distinct types of errors.

All error messages should be printed within the exception handlers in your `main()` function, not within the parser code itself. That is, all references to `cout` and `cerr` should be confined to your `main()` function.

In the case of the "Undefined symbol" error, you **must** somehow print the name of the offending symbol.

Modify your parser so that it no longer relies on return codes to convey error conditions. Depending on how robustly you handled errors in Assignment #2, replacing your old code with exceptions may even simplify your parser.

3 Hints

I will make a solution available for Assignment #2 which can be used as the basis of your Assignment #4 if you wish to use it.

It may be useful to add a `reset()` method to your `Stack` class, in order to guarantee that the stack is in the empty state after an error occurs.

4 Submission

Please email your modified source files, plus any special instructions, to me via email. Good luck! Contact me or your T.A. with any questions or problems.