



University of Glasgow | Department of
Computing Science

FUNCTIONAL PROGRAMMING 4

ASSESSED EXERCISE 2

Calculator

Author:

Garry SHARP
0801585s

Supervisors:

Dr. W. VANDERBAUWHEDE
Dr. J. O'DONNELL

November 30, 2012

Contents

| | | |
|----------|--------------------------------|----------|
| 1 | Instructions | 2 |
| 2 | Status | 3 |
| 2.1 | Windows | 3 |
| 2.2 | The Stack | 3 |
| 3 | Extensions Implemented | 4 |
| 3.1 | Basic | 4 |
| 3.2 | Advanced | 4 |
| 3.2.1 | Nicer GUI | 4 |
| 3.2.2 | Financial Calculator | 4 |
| 3.2.3 | Readable Stack | 4 |
| 3.2.4 | Integer Support | 5 |
| 3.2.5 | Basic Error Handling | 5 |

1 Instructions

Executions is as easy as:

```
$ ghc MainCalc.hs
$ ./MainCalc
```

Please make sure that all files are as the handout files (ie. the .glade file is located at the glade/calculator.glade within the subdirectory you are working in). Also please make sure that each of the steps detailed in the setup.sh file are followed. For quick reference they are:

```
PATH="$HOME/.cabal/bin:$PATH"
PATH="/users/level4/software/fp4/bin/bin:$PATH"
export PATH
HTTP_PROXY="http://wwwcache.dcs.gla.ac.uk:8080"
export HTTP_PROXY
```

As the initial setup, followed by the installation of the following libraries

```
cabal update
cabal install cabal-install
cabal install gtk2hs-buildtools
cabal install gtk
cabal install glade
```

2 Status

The program works and performs calculations as expected, however there are a few minor bugs that I will mention here.

2.1 Windows

The calculator has two additional windows from the standard MainWindow that was given, these are a currency converter window and an about dialog window. These windows will close properly when you click the close button on the lower right section of the panels. Clicking the top right button will also close the window but will not allow it to be re-constructed which is due to the xml file with the hidden xml windows being the one that is read. I did not find a solution to this where I refreshed the xml but this became very convoluted and online forums suggested it was a horrible hack so I removed it. The windows show and hide fine so long as you close them using the close/cancel buttons at the bottom right of the panels.

2.2 The Stack

I am only ever using 1 stack in the code (an array of tuples representing a double and a label string for readability). What I do is when the int mode is selected I change the divide function and floor all of the values and then display the output omitting the decimal point, I think that this is okay as the user will have no inclination to think that the values are still (technically) doubles. Eg a user inputs 45 and this will be stored as 45.0. Int int mode $\frac{45}{7}$ will actually be $\text{floor}\left(\frac{45.0}{7.0}\right)$. The reason for deciding to implement it this way was that it seemed simpler given that we have a binary *"is int mode"* or *"isn't in int mode"* value that we could add to the calculator state to determine HOW we divide values. The alternative of 2 separate stacks or an array of a custom data type seemed both inefficient with memory and too convoluted given the relative simplicity of the extension.

3 Extensions Implemented

3.1 Basic

All of the Basic Extensions are working, there was a level of ambiguity over whether or not the STO button should pop a value from the stack or not, as it does not explicitly say to do so I decided to leave the value there. Therefore, a STO followed by a FET would lead to a stack of, for instance, [2.5, 5.6, 7.8] becoming [2.5, 2.5, 5.6, 7.8].

The about dialog must be closed as detailed in section 2.1

3.2 Advanced

Below are a list of advanced extensions implemented

3.2.1 Nicer GUI

The system's GUI has been redesigned to feature colours which make it clearer which buttons do what, the stack is also printed neatly on one line. A bug in the system is that both integer and double mode are presented to the user at the start of the system. Upon selecting either integer/double mode the GUI fixes itself to how it should be.

3.2.2 Financial Calculator

Clicking convert brings up a new window which allows the user to enter a int the top left, select a currency from and currency to and then convert between them. You must close the window by using the close button in the bottom right as detailed in section 2.1.

3.2.3 Readable Stack

I allow users to also enter a label for a value in the stack and store the values as an array of tuples [(*Double*, *String*)], when the values are printed to the screen the user can see the label they assigned. When a value is pushed onto the stack from an operation (say a multiplication) then the label defaults to "*Auto Gen from*" followed by the name of the operation that created it.

3.2.4 Integer Support

Integers work in the system, when the user clicks the interger radio button a number of operations (sin, cos, inverse etc) dissapear and the mod function appears. As mentioned before in section 2.2, the stack is always a double but when in int mode values are rounded and output as if they were ints.

3.2.5 Basic Error Handling

Basic error handling is performed in the code, please consult comments for instances.