Harvey Mudd College
Computer Science 42
Fall 2009

Assignment 3
Unicalc CLI
Due. 11:59 p.m., Tues., 22 September 2009

This is the second part of a 3-part assignment. **CLI** stands for "Command-Line Interpreter" [or "Interface"] an application that executes user commands from a command line. As before, **Unicalc** is a calculator that includes physical and other units, rather than just numbers.

Use your Unicalc API, or mine, to construct a unit calculator based on a read-eval-print loop. The mechanics of the latter are provided for you in a prototype, **unicalc-scheme3.scm** on the web page, along with a prototype evaluator (which you are free to re-implement) that supplies some of the functionality: *** / let1** and **define**. Examples of input that works with the prototype are on the web in **tests3.uni**. Your task is to define additional functionality:

+	adds two quantities
-	subtracts the second quantity from the first
let	is just like the Scheme let
*application***:**	meaning apply a function to some number of arguments as in (<exp> <arg1> ...) where

<exp>	evaluates to a closure. Typically <exp> is a lambda-expression.
<arg1> …	are evaluated in the current environment.

In addition, we want to allow some abbreviations in user input and output:

> A number by itself stands for the quantity with empty numerator and denominator.

You may add other functionality, but please check with me before going overboard. Also, added functionality must be consistent with what is specified. Examples might include: **let***, **if**, **while**, *etc*.

I will provide a different file **tests5.uni** that should work with your more advanced version.