Homework, Week 1, part B

October 17, 2013

The software development process has four steps. This week's homework will walk you through all four steps. The steps are: (1) analysis, (2) design, (3) implementation, and (4) testing. The process repeats all four steps, one after another, many times before the software is complete.

Implementation The purpose of implementation is to write the software using the design. I have copied a simple addition test program below. Copy this into a file named "add-test.lisp", load it into your clisp environment, and run it.

You do not have to know what this does. This short code has six variables. You should understand what each variable does. They are: count, correct, a, b, c, and d. Do an internet search for "common lisp function defparameter" and "common lisp function let."

```
(defun start-test ()
  (defparameter count 10)
  (defparameter correct 0)
  (format t "Starting the addition test, you have ~a questions.~%" count)
  (run-test))

(defun addition-problem ()
  (let* ((a (random 11))
         (b (random 11))
         (c (+ a b))
         (d (read (format t "What is ~a + ~a? " a b))))
    (cond ((= c d)
           (format t "Correct~%")
            1)
          (t (format t "The answer is ~a~%" c)
              0))))

(defun run-test ()
    (cond
      ((zerop count)
       (format t "you got ~a correct and made a ~a.~%" correct (* 100 (/ correct 10.0))))
      (t (format t "Question ~a. " count)
         (decf count)
         (incf correct (addition-problem))
         (run-test))))
```

Testing Load this file into your lisp with `(load "add-test.lisp")` and evaluate `(start-test)`. Does it do what it is supposed to do? Does it meet all the requirements you identified? Does it do anything that you did not identify as a requirement?