

CSCI 1320 (Principles of Programming I), Spring 2012

Homework 6

Credit: 20 points.

1 Reading

Be sure you have read chapter 8.

2 Programming Problems

Do the following programming problems. You will end up with at least one code file per problem. Submit your program source (and any other needed files) by sending mail to `bmassing@cs.trinity.edu`, with each file as an attachment. Please use a subject line that mentions the course and the assignment (e.g., “csci 1320 homework 1” or “POP I hw6”). You can develop your programs on any system that provides the needed functionality, but I will test them on one of the department’s Linux machines, so you should probably make sure they work in that environment before turning them in.

1. (5 points) In Homework 5 you wrote two programs that compared various ways of operating on arrays and lists in Scala. We left out one way, namely using loops. Your mission for this assignment is to take the programs you wrote for the previous assignment and add to each of them one more method that counts how many times the selected element appears in the array/list. This new method should use a loop.
2. (15 points) Newton’s method for computing the the square root of a non-negative number x starts with an initial guess r_0 and then repeatedly refines it using the formula

$$r_n = (r_{n-1} + (x/r_{n-1}))/2$$

Repetition continues until the absolute value of $(r_n)^2 - x$ is less than some specified threshold value. An easy if not necessarily optimal initial guess is just x .

Write a Scala program that implements this algorithm and compares its results to those obtained with the library function `math.sqrt()`. Have the program prompt for x , the threshold value, and a maximum number of iterations; do the above-described computation; and print the result, the actual number of iterations, and the square root of x as computed using library function `sqrt()`. Also have the program print an error message if the input is negative. Here is a sample execution (with text **in boldface** what you type and text **in typewriter font** what the program prints):

```
enter number to find the square root of:
2
enter threshold:
.00000001
enter maximum number of iterations:
100
result with Newton’s method = 1.4142135623746899 (4 iterations)
result from library function = 1.4142135623730951
```

