CSE 002: Fundamentals of Programming

Spring 2014

Homework Assignment 6: Getting Loopy

Objectives. This homework gives you practice with writing loops.

You may find it helpful to work through the check point questions embedded in the chapter, and to practice on homework problems from the text that we have not ***assigned. Note that solutions to the even-numbered problems are available on the*** *book's student resource website as described on page xii.*

Note that homework is checked for evidence of plagiarism. Academic dishonesty carries a tremendous penalty, and offenders will be caught.

Program 1. I don't usually go to gambling casinos, but I attended the birthday party of a friend at the Mohican Sun, in Connecticut. There I met another person attending the party who told me about how he gambles at roulette. He brings $100 to the table and places a $1 bet on the same (randomly selected) number for 100 spins of the roulette wheel. He claims that he often wins money with this "system." For this program, use a random number generator (recall Math.random()) to simulate this person's betting on 100 spins, but run the simulation 1000 times and collect data on the outcome. (As an alternative, you could buy a roulette wheel and spin it 100,000 times and record the results. Computer scientists prefer to simulate the outcome with a computer. Such simulations are called Monte Carlo simulations.)

Roulette rules for betting on a single number are straightforward: There are 38 possible numbers. If my number comes up I am given $36. For 100 spins, if my number comes up 3 or more times, I walk away with 3x$36=$108 or more. Thus, if my number comes up at least 3 times I win money; otherwise I lose money. For the 1000 repetitions of the 100-spin simulation, compute the number of times I lose everything (my number never comes up), compute the total winnings (of all simulations added together), and compute the number of times I walked away with a profit in one session of 100 spins of the wheel. Store the program in the file Roulette.java and store the file in lab06 on Cloud9.

Program 2. The bisection method is a very simple way to compute the square root of a number. Start by setting low to 0 and high to 1+x. Obviously the square root lies between low (whose square is less than the square root of x) and high (whose square is larger than the square root of x, which is not necessarily true, by the way, for high=x).

Now take the middle of the interval [low, high] and call it middle. If the square of middle is greater than x, then change high to middle; otherwise change low to middle. At this point the square root is still between low and high, but the distance between low and high has been halved. Repeat this process until the difference between high and low is less than some small multiple of 1+x, say 0.0000001*(1+x). Then the square root is very close to both low and high. For example, to find the square root of 2, start with [0,3] (0*0<2 and 3*3 > 2). The middle is 1.5. 1.5*1.5=2.25>2, so the new interval is [0,2.25]. Now the middle is 1.125. 1.125*1.125<2, so the new interval is [1.125,2.25]. And so on. Write a Java program that forces the user to enter a double that is greater than 0, uses the above algorithm to compute the square root of the number, and then prints it out. Use 0.0000001 as the “tolerance” for stopping the loop that computes the square root. Store the program in the file Root.java and store this file in ~/workspace/hw06 on Cloud9.

This homework is due 14 October at 11pm. As usual, late collections will occur on the two following nights.