

CSE 2012 Fundamentals of Computing II

Spring 2008

Lab handout for Week of January 28

Objectives

1. Continue to write C/C++ programs with a focus on classes and elementary data and control structures.
2. Learn a little about Monte Carlo estimation.

Pre-lab assignment

1. Read the Monte Carlo estimation handout written by Prof. Flynn posted on the course web site.
2. (5 points) Submit a C++ class implementation named Point2D (Point2D.cpp and Point2D.h) that stores a single 2D point (x,y). The class should have private slots that store the point's coordinates, and methods that allow the values of the slots to be read and written. The default constructor should assign (0,0) to these slots. A nondefault constructor that specifies the initial x and y coordinates of the point must also be present. A **print()** method should print the contents to **cout**, as an ordered pair (x,y). Finally, provide two utility methods. The first, named **modulus()**, takes no arguments, and returns the Euclidean sum of the x and y coordinates (*i.e.*, $(x^2 + y^2)^{1/2}$) in a double-precision value. The second, named **angle()**, returns the angle in degrees between the x-axis and the line joining the origin to the point (x,y). This angle is given by
$$180 * \text{atan2}(y, x) / 3.1415926535 .$$

Write a test program that initializes an array of ten Point2D objects with random x and y coordinates, and prints out each point's modulus and angle.

3. (5 points) Complete problem 9.11. Include a program that uses a loop to initialize an array of ten Rectangles with random length and width, and uses the class methods to print out the area and perimeter for each one.
4. (5 points) Complete problem 10.8. Hand in a brief report with the new balances calculated by your class as requested in the problem in addition to your code.

In-lab activities

1. (1 point) Report to lab **on time**. Attendance will be taken at the scheduled lab time.
2. Code up a 2D estimator for π that employs N random positions in the 2x2 square centered at the origin. It must prompt for a value of N to be input from the command line. If N>0, it should re-initialize itself and produce an estimate based on N random samples, and then request a new value of N. If N<=0, the program should terminate. Your estimator must use your Point2D class to store the 2D points you generate during the estimation process. **Note**: it might not be all that valuable to keep all of the points around. Hint.
3. Test the estimator until it works. Use N=10000.

4. (4 points) Flag down the lab TA and have them examine and check off your work.

Post-lab (due at the start of next week's lab)

Write and submit (in your dropbox) a lab report with the following sections.

- (5 points) A one-page (or so) description of the problem and your solution.
- (5 points for estimator code only) Code for the Point2D class and your estimator program.
- (5 points) Experimental results. Run your estimator ten times for each of the following values of N: 100, 1000, 10000, 100000, 1000000. Generate a table with five rows and thirteen columns. Each row corresponds to a value of N. The table should contain these items, in the order specified, in each row: value of N, ten values of estimate, mean of estimates, standard deviation of estimates.
- Paste the table into your report.

Code is expected to be well-commented, and the narrative portions of your report must be written professionally (*i.e.*, complete sentences, correct grammar and punctuation, consistent tense and voice, formal tone).