# MATH4900 - Spring 2017

**Dr. Saar Hersonsky**

The project is due 5:00 p.m. on 4.16.

There are five questions - answer 3 of them:one from Q1-Q2, one from Q3-Q4, and Q5.

You may use **only** code that appears in our textbook (chapters 1-2) or in its website (chapters 1-2).

Exception to the above are Java libraries for input/output and animation/drawing.

Quote theorems/propositions/lemmas that you are using in your solutions.

This project is an individual task. In particular, you may not collaborate on any part of it.

Enjoy thinking and coding!

**Q1:**
a) Write a program that generates and displays $N > 1$ random points in the interior of a square of size 10. Divide this into two (or more) methods that will be used in part (b): generating the points and then displaying them in the given square. The number $N$ is read from the terminal.

b) Construct and implement an algorithm that finds a *simple closed polygonal path* that passes through $N$ randomly chosen points in a square of size 10. Your implementation should include: a graphic animation (such as the one we used in the "bouncing balls project" or the "percolation" project) of the algorithm, and an option to read from the terminal the number $N$ and the number of times, $T$, a client may want to test it.

c) What is the complexity of your algorithm? Can you show cases in which it will fail?

**Q2:**

a) (5) Your data is a point $p$ and a *convex* polygon $Q$. Construct and implement an algorithm that decides if $p \in Q$. Your program should have a graphic component that allows the user to view the objects and test your algorithm with various cases.

b) Your data is a point $p$ and a *non-convex* polygon $Q$. Construct and implement an algorithm that decides if $p \in Q$. Your program should have a graphic component that allows the user to view the objects and test your algorithm with various cases.

**Q3:** For any odd, positive number $n$, $3n + 1$ is even, so one can write

$$3n + 1 = 2^k n_1,$$

where $k$ is the highest power of 2 that divides $3n + 1$. Define $f(n) = n_1$. Let $m$ be any odd, positive number which is smaller or equal to 100. Prove that applying $f$ repeatedly to $m$, we always end up with 1. Plot a graph where the horizontal axis presents the number $m$, and the vertical axis presents the number of iterations it takes to end up with 1.

**Q4:**

a) Start with the number 0 and then repeatedly replace 0 with 01 and 1 with 10. So the first four numbers in the sequence are:

$$0, 01, 0110, 01101001.$$

Write a method that produces the $n$-th element of this sequence. Show by a short client program that your method works.

b) Based on this sequence create the following graph. Starting from $(0, 0)$, move one unit in the direction that you are facing if you encounter 1 in your sequence, and rotate by an angle of $-\pi/3$ if you encounter 0 in your sequence. Write a method that produces this graph for the $n$-element in this sequence. Write a short client program that presents a graphic picture of it while reading from the terminal the required $n$ (you are welcome to use scaling of the length of the step for large $n$).

**Q5:** Find the solutions of the equation

$$\sin\left(x^2\right) - \cos\left(x^3\right) = 0,$$

for $0 \leq x \leq \pi$.