Test #1

Test #1 will be on Tue, 20 Oct, during class time and via Canvas.

The test will be published, like a quiz and you have 75 minutes to finish and submit the test. The test may have 4-6 questions and, for each question, there will be a textbox for you to provide the answer. No Multiple choice questions.

The test may consist of questions from the assigned readings, assignments, classwork, and project 1. You should expect code writing questions in Python. You are NOT allowed to use your computing machines to test your code.

It is closed book, but one page of notes (8 1/2 x 11) is allowed. The page of notes can be double sided and printed in any way.

We will NOT use proctoring tools, but you need to join the class and we'll split you into breakout rooms of one person each and you MUST to share your desktop.

It's critical that you join the class at least 5 minutes before it starts to arrange for the breakout rooms.

You should be very familiar with the following:

Extreme Programming Book:

circle of life (Chapter 2): The customer's basic job is to define what has value, and the programmer's job is to build it. Circle of life: define, estimate, choose, build.

stories (Chapter 4): Define requirements with stories, written on note cards. Customer should write these.

acceptance tests (Chapter 5): Acceptance tests should tell the client that the system works, and tell the programmers what needs to get done. "Black box tests"

story estimation (Chapter 6): For each story, assign it a score based on difficulty of implementation. One point should be about one "perfect engineering week" of work. If a story is 3+ points, maybe have the customer split it into two or more stories.

spike solution (chapter 6): Estimating story length when comparing to other stories is easy, but if you haven't implemented anything like this before, do some expereimenting, make a spike solution. Example: balancing an account: take the numbers, add them up. Maybe make a transaction object to store that data, a vector of them, a loop, etc. Formatted reports: make