Introduction to Testing

- Discussion questions
  - What is the goal of testing a software program?
  - During development of a software program, at what point-in-time should we think about testing?

Introduction to Testing (cont'd)

- Discussion questions with answers
  - What is the goal of testing a software program?
    - To find as many errors as we can
    - It's impossible to mathematically prove a program is correct
  - During development of a software program, at what point-in-time should we think about testing?
    - As soon as we begin to understand the problem statement
    - Testing should be a part of every development step
    - We iteratively develop an entire solution
      - Repeat until done:
        - Write some code
        - Test your code

Two (basic) Types of Testing

- Black Box
  - Test based on problem statement
    - i.e., test ***what*** needs to be done
- White Box
  - Test based on program design & code
    - i.e., test ***how*** it is done

The Process of Testing

(Option 1: Test Cases)

- Document test cases
- What's a test case?
  - In simplest form, describe input data and expected result
  - Document lots of test cases to try to discover as many logic errors as possible
- Use these test cases when executing your solution
  - Document the actual results
  - Do the actual results match the expected results?

The Process of Testing

(Option 2: Test Code)

- Write test code
- What's test code?
  - Code that is not part of your solution
  - Code that exercises your solution to try to discover as many logic errors as possible
    - i.e., Your test code implements test cases (that you may have documented)
- Each time you execute your solution, you are running your test code
  - Test code will typically display results
    - So you can detect logic errors

The Process of Testing

(Other Options)

- There are tools that automate testing
- There are different kinds of testing
  - Unit testing
    - Test individual software elements (e.g., each method)
    - Done by programmer
  - Integration testing
    - Test a bunch of software elements (e.g., a bunch of classes)
    - Typically done by programmer
  - System testing
    - Test entire software application/system
    - Done by programmer, analyst or a separate testing team
  - Acceptance testing
    - Test performed by customer/client

The Process of Testing
(Black Box)

- Thinking about (black box) testing ***early*** is beneficial
  - You gain a better understanding of requirements (i.e., problem statement)
  - You begin to develop ideas about what your solution needs to do
    - This may be
      - Specific details that you discover should be in your solution
        - e.g., methods needed
      - General design ideas
        - e.g., classes needed, relationships between classes

The Process of Testing
(White Box)

- Thinking about (white box) testing is a great way to confirm your understanding of your code
  - You may discover logic errors as you develop white box test cases (or test code)

The Process of Testing

(Minimum Testing Guidelines)

- Black Box
  - Test each requirement at least once
- White Box
  - Ensure each statement is executed at least once
    - When you have complex selection logic, this can dramatically increase the number of test cases needed
  - Ensure all boundaries of each simple condition are tested
    - i.e., this requires (at least) two distinct test cases
    - When you have complex Boolean expressions, this can dramatically increase the number of test cases needed

The Process of Testing

(For Assignment 5)

- String class
  - length() method
    - Returns number of characters in the String object
- Doing validation in a constructor method
  - Allow object to be constructed
    - Generally bad idea to *terminate* a constructor
  - If validation logic finds invalid value
    - Initialize corresponding instance variable to value that denotes invalid data found
- Use of super(…)
  - See chapter 11 in text book

The Process of Testing

(For Assignment 5 cont'd)

- Scenario
  - Define a class named `aNumber` that has:
    - One instance variable
      `int number`
    - One constructor
      `public aNumber(int number)`
      - `number` formal parameter variable must in range [-100,100].
    - Three methods
      - `public void increment()` `//adds 1 to number`
      - `public void decrement()` `//subtracts 1 from number`
      - `public int get()` `//returns value of number`
  - What does code:
    - For `aNumber` class look like?
    - That tests constructor method look like?