

## CS 2510 Exam 1 Cover Page – Spring 2017

Problem	Points / Possible
1	/ ???
2	/ ???
...	/ ???
<i>n</i>	/ ???
<b>Total</b>	/ ???

Name: \_\_\_\_\_

Student Id (last 4 digits): \_\_\_\_\_

Instructor: \_\_\_\_\_

Lecture section (time): \_\_\_\_\_

- You may use any of the Java techniques we have learned so far.
- Some basic test taking advice: Before you start answering any problems, read *every* problem, so your brain can be thinking about the harder problems in background while you knock off the easy ones. ***The sub-parts of each problem are related; this may be of some help...***
- Remember that the phrase “design a class” or “design a method” means more than just providing a definition. It means to design them according to the **design recipe**.
  - ⇒ If an interface is given to you, you do not need to repeat the signature and purpose statements in your implementations.
  - ⇒ Likewise, you do not need to repeat any test cases given to you, but you should add tests wherever appropriate.
  - ⇒ You **do not** have to write out the template explicitly for each method or class. However, if you are stuck on how to proceed, writing the template may help you. And, indicating which items in the template you think are useful for a given problem may help you earn partial credit.
- We will not answer *any* questions during the exam, except possibly to fix typos or errors that we may have overlooked.
- The following shorthand notations can be used:
  - ⇒ To write test cases, you may use `c → e` instead of `t.checkExpect(c, e)`. For example, you may write `1 + 1 → 2` instead of writing `t.checkExpect(1 + 1, 2)`.
  - ⇒ To construct `ILists` *in your tests*, you may write `makeList(item1, ..., itemN)`. For example, you may write `makeList(2, 5, 1, 0)` to make a list of integers. In your code, use the actual constructors as needed.
- To add a method to an existing class definition, you may write just the method and indicate the appropriate class name rather than copy the entire class definition.

***Good luck!***