Name: ____________________________________ Section: ____________

CSSE 220—Object-Oriented Software Development

Exam 2 – Part 2, Oct. 21, 2016

Allowed Resources on Part 2. Open book, open notes, and computer. Limited network access. You may use the network only to access your own files, the course Moodle and Piazza sites (but obviously don't post on Piazza) and web pages, the textbook's site, Oracle's Java website, and Logan Library's online books.

Instructions. *You must disable Microsoft Lync, IM, email, and other such communication programs before beginning part 2 of the exam. Any communication with anyone other than the instructor or a TA during the exam may result in a failing grade for the course.*

You must actually get these problems working on your computer. Almost all of the credit for the problems will be for code that actually works. There are several different small methods to write, so you can get a lot of partial credit by getting some of them to work. If you get every part working, comments are not required. If you do not get a method to work, comments may help me to understand enough so I can give you (a small amount of) partial credit.

Begin part 2 by checking out the project named *Exam2-201620* from your course SVN repository. (Ask for help immediately if you are unable to do this.)

When you have finished a problem, and more frequently if you wish, **submit your code by committing it to your SVN repository**. We will check commit logs, so you must be careful not to commit anything after the end of the exam. For grading, we will ensure that the included JUnit tests have not been changed.

Part 2 is included in this document. **Do not use non–approved websites like search engines (Google) or any website other than those mentioned above.** Be sure to turn in these instructions, with your name written above, to your exam proctor. You should not exit the examination room with these instructions.

Problem Descriptions

Part C1: Recursion Problems (21 points)

The class Recursion in the recursion package contains 4 recursion problems (test cases are also included). *You only need to solve 3 of the 4 problems.* Leave the problem you chose to skip blank and leave a comment saying that you skipped it. These problems must be solved with recursion - **a working solution with loops is worth no credit**. If you have time and want to do a fourth one for fun, that's fine, but we suggest saving it until you finish the rest of the exam.

Part C2: Polymorphism Problem (6 points)

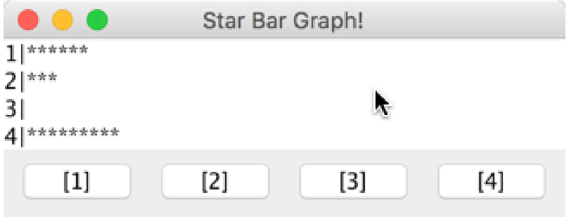
You are given a working solution to a program that computes discounts on food. But sadly, there is code duplication: two methods to compute discounts because there are two kinds of discounts.

You can run it either using a DiscountMain class, and try it out with a text-based interface, or using the DiscountMainTest unit tests. It currently passes the tests.

Your job is to create an interface and then use it to remove the code duplication, so that you only have one method to compute discounts, not two. Make any other small changes you need to the code to make it work with your new interface. The javadoc in DiscountMain has more details.

Your solution must use an interface, remove the code duplication, and pass all the unit tests to get credit.

Part C3: Star Bar Graph (17 points)



```
1|******
2|***
3|
4|*********
```

Screenshot of the completed program.

In the package starBarGraph is code for a GUI framework for a simple program that displays a bar graph made of stars (asterisks). You need to add the buttons, listeners, and variables necessary to make it interactive. You may add any new classes or make any changes you feel necessary.

Stage 1 (5 points) Four buttons, labeled 1-4, appear on the window as shown in the image above. You may add any other visual elements necessary to accomplish this.

Stage 2 (12 points) Clicking a button adds a star to the appropriate graph. Add the ActionListeners and other items necessary to get the application working. You may add new methods, classes, ActionListeners, etc. Using static variables to access data is bad practice, so **if you use static variables - yuck!**, you will earn a maximum of 6 points on this part. Furthermore, the code for all 4 buttons is pretty similar. If you create four different, nearly-identical listeners **(more code duplication - double yuck!)** then you will earn a maximum of 9 points on this part.

Note that the given code shows some stars. Your graphs can start with the given amount of starts, or you can start the program with all the stars being 0.