

CSSE 220—Object-Oriented Software Development

Exam 2 – Part 2, October 20th, 2017

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 *Exam2-201810* 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 (18 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 (8 points)

You are given a working solution to a program that draws people with various hats. Unfortunately this code has a lot of duplication both in the FaceComponent and in the various Face classes. Your job is to use inheritance to remove the code duplication. Make any other small changes you need to the code to make it work with your modified classes.

You can run it by running the FaceComponent. Your changes should not affect the functionality of the solution (i.e. it should keep drawing exactly the same picture it draws right now).

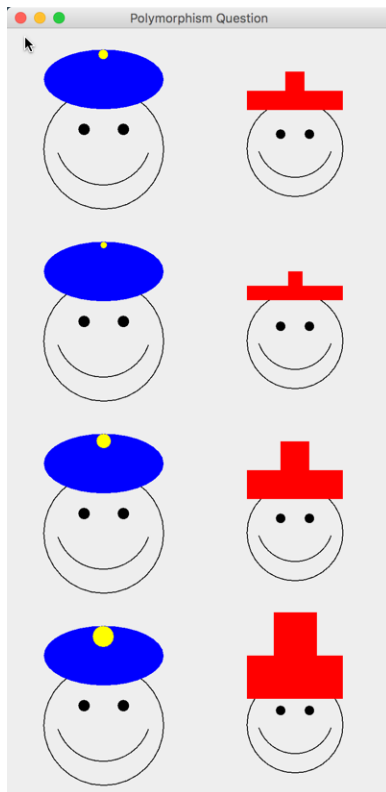


Figure 1: Picture

Part C3: Clock (18 points)

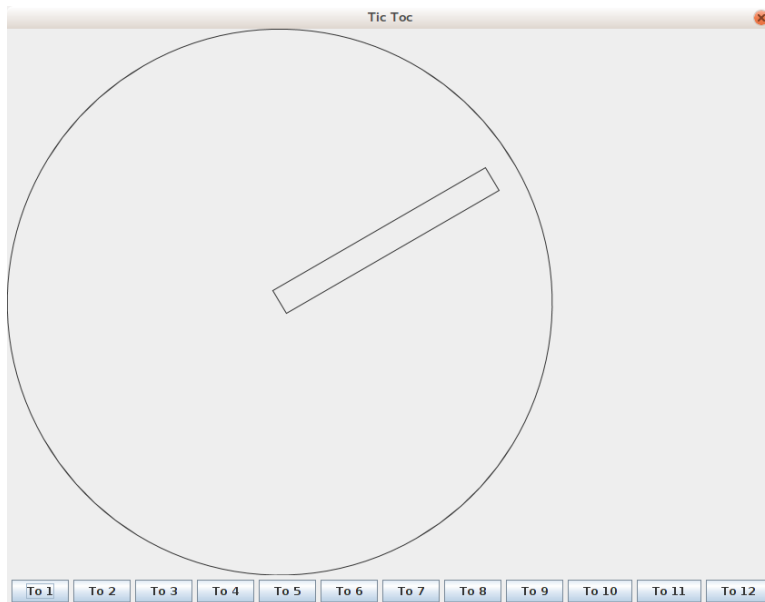


Figure 2: Final UI

In the package `events` is code for a simple program that displays draws a clock at various hours. The drawing code has already been written for you, but you will have to add the buttons and make them work.

You may add any new classes or make any changes to existing code you feel necessary.

UI: (7 points) Add buttons to the window so it matches the diagram.

Buttons that do something: (5 points) Add listeners to the buttons that print a message to the console when the buttons are pressed. So when button “To 1” is pressed, the message “Button 1 Pressed” should appear in the console.

You should do this without tons of code duplication (e.g. don’t have a `Button1Listener.java` `Button2Listener.java`, etc.)

Buttons update the clock: (6 points) Make the buttons update the clock. To do this, you’ll want to add a method that lets you set a time to `HourTimerComponent`. Be sure to call `repaint` (inherited from `JComponent`) to trigger `paintComponent` being called again.

Using static variables will lose you half of the credit on this part.