

# CSSE 220—Object-Oriented Software Development

## Exam 1 – Part 2, September 21, 2018

**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 Skype, 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 zip from from Moodle and opening it in eclipse.** (Ask for help immediately if you are unable to do this.)

*Part 2 is included in this document. Do not use non-approved websites like search engines (Google) or any website other than those above.* (Exception: you may use a search engine to search for Java documentation on Oracle's java website.) Be sure to turn in the these instructions, with your name written above, to your exam proctor. You should not exit the examination room with these instructions.

---

### Honesty pledge.

I understand that I may not communicate in any way with anyone other than the instructor and their assistants or use any non-approved resources during the exam.

I understand that after the exam, I will not communicate anything about the exam to any student that has not already taken the exam.

I understand that if I violate either of the above, that the penalty is at least a -100% on this whole exam, and that I may be expelled from Rose-Hulman.

If you understand these and agree to abide by them, then check here: \_\_\_\_\_

Otherwise, check here and talk to your professor privately soon after the exam: \_\_\_\_\_

Your name (print legibly): \_\_\_\_\_

## Part 2—Computer Part

---

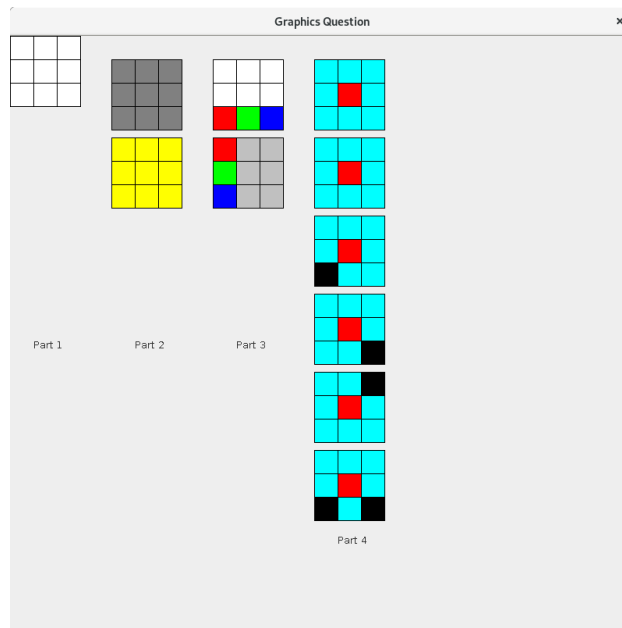
### Problem Descriptions

**Part A: Small Problems** (20 points) Implement the code for the 2 functions in `SmallProblems.java` – each problem is worth 10 points. Instructions are included in the comments of each function. Unit tests are included in `SmallProblemsTest.java`.

**Part B: Map and 2D Array Problems** (20 points) Implement the code for both of the functions in `MapAnd2dArray.java` – each problem is worth 10 points. Instructions are included in the comments of each function. Unit tests are included in `MapAnd2DArrayTest.java`.

**Part C: Test This Class** (7 points) Implement a unit test for the function in `TestThisClass.java`. You will add a file `TestThisClassTest.java` that will contain your test. Your test should have 3 assertions that test a variety of cases, but need not be exhaustive.

**Part D on next page**



The final output

#### Part D: ThreeByThree (18 points)

Read over all these instructions carefully. Make sure you understand completely what functionality you have to implement before you start coding. Ask questions if anything is unclear.

- Phase 1 (4 points) When constructed with no parameters, ThreeByThree should draw a 3x3 grid of squares with an upper left corner at 0,0. The squares should be 30 pixels (SQUARE\_WIDTH) on a side filled white with a black border.
- Phase 2 (4 points) Uncomment the code in ThreeByThreeComponent.java and add a new constructor that takes 3 parameters. The first two parameters should be the x y coordinates of the ThreeByThree's upper left corner. The third parameter should be a Color which should be the background color of the squares.
- Phase 3 (7 points) Uncomment the code in ThreeByThreeComponent.java and add a new method setCellColor to ThreeByThree which takes three parameters. The first two parameters should be a row number and column number (0 indexed) which indicate a particular cell in the ThreeByThree, the third parameter is a color. setCellColor should make it so when that cell is drawn, it has the given color rather than the background color for its ThreeByThree.
- Be sure your picture matches the given one.
- Phase 4 (3 points) Uncomment the code in ThreeByThreeComponent.java and add a new method copyTo to ThreeByThree. This function should return a new ThreeByThree which is a copy of the original but with a different x and y coordinate (passed as parameters to copyTo). The copy should have the same background color and cell colors as its source at the time of the copy. It should be possible to use setCellColor to further modify the copy before it draws, and those modifications should not affect the source.

Be sure your picture matches the given one.