

# CSSE 374: Exam 2

---

## Policy

This is an open-book, open-note, open-moodle, take-home exam. You are also allowed to refer to any documents on the internet provided that you are not communicating or collaborating with another individual. Asking questions in online-forums, newsgroups, or via chat applications and emails are also forms of collaboration and are not allowed in this exam.

This part of the exam contributes 80% to your Exam 2 grade and 8% to your overall course grade. The exam is due by 5:10 pm, Friday, 1/23/2015 on Moodle.

## Problem

All of you are awesome students as you are studying in #1 engineering school! If you feel otherwise 😊, assume that you are an inquisitive CSSE student, passionate about learning new stuffs, and sometimes, you design and code things just for the fun of it! Chandan asked you to develop something which may seem utterly useless at first, but can have useful applications. So, you decided to be a sport and go with it. Here is the conversation between you and Chandan assuming that your name starts with D.

**C:** D, can you code up the following example <http://docs.oracle.com/javase/tutorial/uiswing/painting/step2.html?>

**D:** Yes, I can! I created a file named **problem.SwingPaintDemo2** and when I run it, I can see the text on the panel.

**C:** Great job! Can you also draw some graphical shapes; try a rectangle and a circle? You can draw them at any reasonable locations and can give each of them a reasonable size.

**D:** All done! I have carefully reviewed all of your solutions and I know that the solution for Lab3-1 has examples on how to draw shapes using the graphics objects.

**C:** Wonderful! Now that you are all warmed up, do you want to try something more interesting?

**D:** Yes! Yes! Let's do it!

**C:** Ok, here is the problem:

I would like you to design a simple drawing application. It should start with an empty **JPanel**. When you press the "r" key, your app should draw a rectangle with a default size (you choose) at a reasonable random location on the panel. You will need a **KeyListener** for this to work. Keep in mind that you can add a key listener to any JComponent such as JFrame, JPanel, JButton, etc. For the key press to work, the component that has the key listener should have the focus. Here are two examples for your reference:

- <http://docs.oracle.com/javase/tutorial/uiswing/events/keylistener.html>
- [http://www.java2s.com/Tutorial/Java/0260\\_Swing-Event/HowtoWriteaKeyListener.htm](http://www.java2s.com/Tutorial/Java/0260_Swing-Event/HowtoWriteaKeyListener.htm)

I would like you to implement the following key-press behaviors in your application:

- **r** => Draw a **R**ectangle at a random location with the default size
- **c** => Draw a **C**ircle at a random location with the default size
- **s** => **S**tore the shapes in the file system. Use the **input\_output** directory.
- **t** => **R**estore the shapes from previously stored state in the **input\_output** directory if one exists.
- **u** => **U**ndo the last action. Your application should be able to maintain a history so that you can undo more than once. You don't need to support redo.

I hope you realize how important this application can be. So, I would like you to design it such that it allows easy addition of new keys/behaviors in the future and has low maintenance overhead.

**D:** Piece of cake! All Done!

**C:** Awesome! Keep up the great work!

## Design

Create **Exam2/docs/Answer.pdf** with answers to the following problems:

**Q1.** Create a UML Class Diagram to present your design idea and explain it in a few lines. Also identify the design patterns you used in your design. **[20 points]**

## Implementation

**Q2.** Implement your solution in the **Exam2/src/problem** package. **[50 points]**

## Testing

**Q3:** Implement necessary test cases in the **Exam2/test/problem** package that tests all of the important aspects of the drawing application. **[10 points]**