

<b>Name:</b> <i>(as it would appear on official course roster)</i>		
<b>Umail address:</b>	@umail.ucsb.edu	section 4, 5, or 6
Optional: name you wish to be called if different from name above.		
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")		

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

# 1

# H08

CS56 W16

## H08: Due Tuesday, 01.19 in GradeScope

GUIs, Graphics (HFJ Ch12, pp 353-355, 363-368 (skimming 356-362, deferring 369-398))

Assigned: Mon 01.11      Total Points: 50

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE,  
OR IF APPLICABLE, SUBMITTED ON GRADESCOPE. There is NO MAKEUP for missed assignments;  
in place of that, we drop the five lowest scores (if you have zeros, those are the five lowest scores.)

### Reading Assignment:

- Read just the first part of HFJ:Chapter\_12, starting on p. 353, and ending on p. 368.
- Also read HFJ:Chapter\_17, pp. 581-595, about Jar files and Packages.
- You can skip the stuff about "Java Web Start"—that's a technology no longer widely used.

### You need this reading for upcoming labs work

- Ch. 12 is titled Getting GUI: A very graphic story, and we need these concepts for an upcoming lab:
  - How to create a JPanel 353-355.
  - How to draw 2-D graphics 363-368.
  - The material on pages 356-362 is about how to work with widgets, events, and ActionListeners, and we won't need that until later.
  - So, if you want, you can just skim over 356-368 for now; that will be covered on a future homework assignment.

1. (4 pts) Fill in the homework header properly—this helps us keep the grading pipeline flowing so that you get credit for your work and get feedback more quickly.

- writing **either 4, 5, or 6** to indicate your **discussion section (lab)** meeting time
- entering BOTH your name AND your umail address EVERY time.

**Paper submissions:** One sheet of 8.5x11 paper double sided, or two DISCONNECTED SHEETS with your name on EACH. Please: **NO STAPLES, NO PAPERCLIPS, NO TAPE, NO ATTACHMENT OF ANY KIND.** These damage the document scanner.

**Scanned submission:** When submitting by PDF upload: scan your pages legibly and **SCAN IN THE CORRECT ORDER.** Page 1 first, then Page 2, in the correct orientation. Failure to scan properly may result in zero credit, meaning you "use up" one of your five "drop the lowest grade" slots.

2. (5 pts) What is the name of the Java class for an object that represents a window on the screen?

3. (6 pts) The text says: "If you want to put your own graphics on the screen, your best bet is to make your own paintable widget." The book recommends that to do this, you should: (fill in the blanks):

Make a \_\_\_\_\_ of JPanel,

and override one method: \_\_\_\_\_

4. Now, several questions about this "mysterious method", i.e. the answer to the 2nd blank in the previous question. (The method you override.)
- (5 pts) The book says that you write code to go inside this mysterious method when you override it. What kind of code do you write inside this mystery method? (Note: pp. 363-368 have lots of examples.)
  - (5 pts) The book says that you never call this mysterious method yourself—and the reason is that the parameter to that method is something you don't have direct access to. What is the Object that is the parameter to the mysterious method, that the System has access to, but you as a programmer don't?
  - (5 pts) Though you can't call the mysterious method directly, there is a method that you **can** call that *asks* the system to call that method for you. What is this method that you **can** call?
5. Inside this mysterious method, one often finds this mysterious line of code.
- ```
Graphics2D g2d = (Graphics2D) g;
```
- (5 pts) Explain **WHAT** this line of code is doing
  - (5 pts) More importantly, Explain **WHY** that may be needed or helpful
6. (10 pts) Write the code for a paintComponent method that fills a square of a random color on the screen, with side length 100, and **CENTERED** at 50,200. (Hint: You'll need to look up the parameters to fillRect in the Java API online.)