

First name (color-in initial)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	section (10, or 11)	first name initial	last name initial
Last name (color-in initial)	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z			

H11: Due Wednesday 05.01 in Lab. Total Points: 50

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

MAY ONLY BE TURNED IN DURING Lab ON Wednesday 05.01, or offered in person, for in person grading, during instructor or TAs office hours.

See the course syllabus at <https://foo.cs.ucsb.edu/56wiki/index.php/S13:Syllabus> for more details.

(1) (10 pts) Fill in the information below. Also, fill in the A-Z header by

- **coloring in** the first letter of your first and last name (as it would appear in Gauchospace),
- writing **either 10,11** to indicate your **discussion section** meeting time
- writing your **first and last initial** in large capital letters.

All of this helps us to manage the avalanche of paper that results from the daily homework.

name:	
email address:	@umail.ucsb.edu

Reading Assignment:

- Read just the first part of HFJ:Chapter_12, starting on p. **353**, and ending on p. 368.
- Earlier in the quarter we read the first part of HFJ:Chapter_17, pp. 581-595, about Jar files and Packages. Review those again.

You need this reading for lab05

- Ch. 12 is titled Getting GUI: A very graphic story, and we need these concepts for this week's 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, and come back to that when you do the reading for H12.
 - Material on p. 369-398 will also be covered on H12.

(2) (4 pts) What is the name of the Java class for an object that represents a window on the screen?

(3) The text says: "If you want to put your own graphics on the screen, your best bet is to make your own paintable widget."

(a) (6 pts) 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.)

(a) (4 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.)

(b) (4 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?

(c) (4 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?

(d) Inside this mysterious method, one often finds this mysterious line of code.

```
Graphics2D g2d = (Graphics2D) g;
```

- (4 pts) Explain WHAT this line of code is doing
- (4 pts) more importantly, Explain WHY that may be needed or helpful

(5) (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.)