

4. (3 points) Which of the following is the least related to the view hierarchy?
 - a. Input handling
 - b. Clipping
 - c. Models
 - d. Automatic layout

5. (4 points) Give 2 reasons why the view and the controller are often tightly coupled.

6. (4 points) List 4 kinds of translated input events.

7. (4 points) The observer pattern is typically used in MVC as a way for the controller to receive input events. How else is the observer pattern used in MVC?

8. (4 points) Louis Reasoner wants to use mouse triple-clicks in his application. He decides to measure the time between clicks by checking the system clock with `System.currentTimeMillis()`. What's wrong with this approach?

9. (4 points) Alyssa Hacker is designing a checkers game in which the user makes moves by dragging checkers around the board with the mouse. She's wrestling with a strange bug: if the user is dragging a checker and moves the mouse quickly, the checker sometimes stops following the mouse, even though the user hasn't released the mouse button. The checker freezes in place until the user moves the mouse pointer back over it, at which point it starts moving with the mouse again. (You don't know what UI toolkit Alyssa is using, but it isn't Swing.)

From this behavior, describe:

- a. the output model she's using to represent the checkers;
- b. how she's handling the mouse input for checker dragging;
- c. what's causing the bug, and how it might be fixed.

10. (4 points) Give one advantage and one disadvantage of using components for output, as compared to strokes.

11. (4 points) Louis Reasoner is writing a Swing widget. He figures he can improve its responsiveness by saving the Graphics object passed to paint(), so that he can draw to it immediately whenever the model changes. Here's some of his code:

```
public class LouisWidget extends JComponent {
    Graphics savedGraphics;
    ...

    void paint (Graphics g) {
        savedGraphics = g; // save Graphics object for later use

        g.drawLine (...); // draw widget as usual
        ...
    }

    void modelChanged () {
        if (savedGraphics != null) {
            // draw changes on the screen right away
            savedGraphics.drawLine (...);
            ...
        }
    }
}
```

Although Ben Bitdiddle thinks Louis's optimization is unnecessary, he's more concerned that Louis has implemented it incorrectly.

What is it about the Graphics object – and analogous objects in other UI toolkits – that makes it a bad idea to save it like this?

12. (3 points) In the HSV color model, what color is the fully unsaturated counterpart of pure red (i.e., identical on all components except $S = 0$)?
- Black
 - White
 - Pink
 - Green
13. (3 points) RGB and HSV color models are both three-dimensional – i.e., each color has three components. RGB is represented by a cube, logically enough. But HSV isn't. How is HSV represented, and why?
14. (4 points) Give 2 reasons to use your platform's native button widget rather than implementing your own -- 1 reason related to software engineering, and 1 reason related to usability.
15. (3 points) List 3 kinds of user tests.
16. (3 points) The double-buffering technique is used for:
- Coalescing mouse movements
 - Preventing screen flicker
 - Capturing the mouse
 - Transforming coordinate systems

17. (4 points) In this problem, you will implement an output effect called a **drop shadow**. A drop shadow makes it look like a screen object is casting a shadow by drawing an identical copy of the object shifted slightly in both x and y directions. Here are some examples of drop shadows:



For the purposes of this problem, a drop shadow is **gray** and shifted **4 pixels** in the positive x and y direction.

Create a drawing surface `ShadowGraphics` that automatically draws a drop shadow for every drawn stroke. `ShadowGraphics` is a wrapper around a `Graphics` object that does the actual drawing. Here's an example of using `ShadowGraphics`:

```
void paint (Graphics g) {  
    ShadowGraphics sg = new ShadowGraphics (g);  
    sg.drawText (0, 0, "This text has a drop shadow");  
    g.drawText (0, 50, "This text has no shadow");  
}
```

Here's what might appear on the screen for this code:

This text has a drop shadow
This text has no shadow

For this problem, we'll use a simplified version of `Graphics` that only draws text:

```
interface Graphics {  
    void drawText (int x, int y, String text);  
  
    void setColor (Color color);  
    Color getColor ();  
  
    void setFont (Font font);  
    Font getFont ();  
}
```

So you don't need to worry about lines, rectangles, circles, etc.

Fill in the skeleton of `ShadowGraphics` on the next page so that it automatically draws a drop shadow for every `drawText()` drawing call.

```
class ShadowGraphics implements Graphics {
    Graphics g;

    ShadowGraphics (Graphics g) {
        this.g = g;
    }

    void drawText(int x, int y, String text) {

    }

    void setColor (Color color) {

    }

    Color getColor () {

    }

    void setFont (Font font) {

    }

    Font getFont () {

    }
}
```

18. (4 points) Ben Bitdiddle proposes implementing drop shadows for the component model: a container class `ShadowContainer` that draws drop shadows automatically for all the components inside it. Describe how `ShadowContainer` could be implemented using `ShadowGraphics`.

Suppose you're a facilitator for a user test of a computer prototype. What's wrong with each of the following situations, and what should you do to fix or prevent the problem?

19. (4 points) When the user starts the first task, you discover that the prototype isn't broad enough to cover it.
20. (4 points) During the test, the room is totally quiet.
21. (4 points) After the test, the user says, "I really felt like an idiot making all those mistakes. I hope you weren't recording me with that video camera."

You're trying to use a drawing program on your workstation, but Ben Bitdiddle is logged on to your computer running a huge job in the background, taking up 99% of the CPU. You notice it most when you try to draw a freehand scribble with the mouse.

22. (4 points) If mouse moves are **coalesced** in the event queue, what effect do you see when you try to draw a freehand stroke?

23. (4 points) If mouse moves are **not coalesced** in the event queue, what effect do you see when you try to draw a freehand stroke?

24. (3 points) Which of the following statements is **not** true about the UI toolkit described in the paper "Glyphs: Flyweight Objects for User Interfaces"?

- Individual text characters are represented as components.
- Glyphs support automatic layout.
- A glyph must store its screen position.
- Glyphs are components (as opposed to strokes or pixels).

Ben Bitdiddle is designing an experiment comparing handwriting recognition with keyboard text entry.

25. (3 points) The experiment will measure text entry speed and error rate. These measurements are:
- independent variables
 - dependent variables
 - uncontrolled variables
26. (3 points) The experiment uses a between-subjects design. This decision threatens:
- internal validity
 - external validity
 - reliability
27. (3 points) Handwriting recognition will be tested on a Tablet PC, while keyboard entry will be tested on a desktop PC. This decision threatens:
- internal validity
 - external validity
 - reliability
28. (3 points) For each trial of the experiment, users will enter the letters of the alphabet in order. This decision threatens:
- internal validity
 - external validity
 - reliability