

Exam 2 Sample Problems

CS 1101B – Fall 2014

Not using the computer:

1. What's wrong with the following `while` loop that is intended to print out, for every number from 1 to 100, inclusive, whether that number is even or odd?

```
int counter = 0;
while {counter > 100} {
    if (counter % 2 == 1)
        System.out.println(counter + " is odd.");
    else
        System.out.println(counter + " is odd.");
    counter++;
}
```

2. Describe the output produced by this `while` loop:

```
int k = 5;
int i = -2;
while (i <= k) {
    i = i + 2;
    k--;
    System.out.println(i + k);
}
```

3. Use nested `while` loops to print out a multiplication table for 1 through 3, i.e.:

	1	2	3
1	1	2	3
2	2	4	6
3	3	6	9

4. Global (or instance) variables, local variables, and method parameters are all used to hold data. Explain the kind of data each one holds and under what circumstances that particular way of holding data would be necessary (or desirable).

Using the computer:

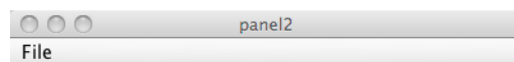
1. Write an `ActiveObject` class called `RisingSun` that creates a sun that rises and sets. It should not be visible when it is created, but then it should slowly rise, gradually appearing above the bottom of the canvas, and rise until it touches the top of the canvas. It should then start setting, sinking until it is completely below the bottom of the canvas. The `x` coordinate of the center of the sun when it is created and the

size of the sun should be values that are passed in to the constructor. The sun should start out orange (`new Color(255, 79, 0)`) and slowly turn yellow as it rises. The color change should reverse itself on the way down. Test your `RisingSun` class by writing an `Events` class that creates a `RisingSun` whenever the user presses the mouse in the canvas. The `x` coordinate of the center of the `RisingSun` created should be the same as the `x` coordinate of the point where the user pressed the mouse. Note, however, that no matter what the `y`-coordinate is where they pressed, the sun should start below the horizon.

2. Write a `WaterGlass` class that creates a simple graphical depiction of a glass with some amount of water in it (from none to full) and that can change the level of the water. Make the glass a framed rectangle and the water a blue (actually, CYAN looks nicer), filled rectangle of the appropriate dimensions at the “bottom” of the framed rectangle that is the glass. Your class should have the following methods:
 - (a) a `WaterGlass` constructor that creates an empty glass. Both the location of the upper-left corner of the glass and the dimensions of the glass are specified by the call to the constructor.
 - (b) a `setWaterLevel` method that sets the water level such that the top of the water is at the `y`-coordinate specified by the call to the method (see the note below),
 - (c) a `contains` method that returns `true` if the location sent to the method is in the glass; `false` otherwise, and
 - (d) an `emptyGlass` method that “empties” the glass (not gradually, but all at once).

Note: You must not create more objects than you need. In particular, you must not repeatedly create new filled rectangles when you change the water level.

Here is an example that shows the glass when the glass is partially full.



I have provided a complete `Events` class that tests the `WaterGlass` class and shows how its methods are used. **You should not change this class.** The comments in that class explain what it does.