

Computer Science 51
Spring 2006

Midterm Examination
8 March, 2006

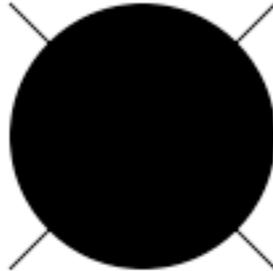
Question	Points	Score	Description
1	15		Class definition
2	10		Program analysis
3	15		Programming - Swing
4	13		Short Answers
Total	53		

This examination is closed book. You have 50 minutes to complete the exam.

Your Name (Please print) _____

1. **Classes:**

I would like you to design a class representing a draggable “Sun-like” object that should look like this (except for the color):



You can think of this “sun” as having four rays emanating out (though you can draw them with just two lines, if you think about it a bit). The constructor should allow the sun to be created at any location and of any size(though of course it should be round!). It should always be yellow when it is created. Your sun should be “draggable,” so it will need a `move` method and a `contains` method (but no `moveTo` method).

To make things slightly more interesting, the sun’s rays should disappear and the sun should always change color to orange when it is being moved. The method `shineOn`, should make the rays reappear and restores the sun to its usual yellow color.

You are to fill in the class below to implement this sun class. Fill in any needed constants, instance variables, local variables, formal parameters, and method bodies. You need not include comments.

```
public class Sun {  
  
    public Sun(  
  
    ) {  
  
  
  
  
  
  
  
  
  
}
```

```
// move the sun, hide the rays, and change color to orange
public void move(                                     ) {

}

// show the rays and change color to yellow
public void shineOn() {

}

// determine whether sun (including rays) contains the point
public boolean contains(Location point) {

}

}
```



```
import java.awt.*;
import objectdraw.*;

public class Mystery extends WindowController
{
    private static final Location textPlace =
        new Location(150,450);
    private MysteryObject thing;

    public void begin() {
        new Text("hold the mouse down for a while",
            textPlace,canvas);
    }

    public void onMousePress(Location point) {
        thing = new MysteryObject(point,canvas);
    }

    public void onMouseRelease(Location point) {
        thing.release();
    }
}
```

```

import objectdraw.*;

public class MysteryObject extends ActiveObject {
    private static final int START_DIAMETER = 4;
    private static final double INCREASE = 2;
    private static final int PAUSE_TIME = 50;

    private FramedOval circle;
    private boolean growing = true;

    public MysteryObject(Location point,
                          DrawingCanvas canvas)
    {
        circle = new FramedOval(
            point.getX()-START_DIAMETER/2,
            point.getY()-START_DIAMETER/2,

            START_DIAMETER,START_DIAMETER,canvas);
        start();
    }

    public void release() {
        growing = false;
    }

    public void run() {
        while (growing) {
            circle.setWidth(circle.getWidth()+INCREASE);
            circle.setHeight(circle.getHeight() + INCREASE);
            circle.move(-INCREASE/2,-INCREASE/2);
            pause(PAUSE_TIME);
        }

        while(circle.getY() + circle.getHeight() >= 0) {
            circle.move(0,-INCREASE);
            pause(PAUSE_TIME);
        }
    }
}

```

3. A very enthusiastic CS 51 student named Herb became so excited when he learned about GUI components that he wanted to go back and revise the programs he wrote for some of the earlier labs to use them. For example, Herb wanted to change the Frogger program so that the player would have to click on appropriate buttons to move the frog rather than just clicking on the canvas. Unfortunately, despite his enthusiasm, Herb doesn't really understand how to write programs using GUI components. So, we would like you to help him get started on the task of revising his Frogger by writing a program to illustrate how a set of buttons could control the hopping of the frog.

We don't want you to write all the code Herb will need. After all, this should be a learning experience for Herb. Instead, we just want you to write a simple program that will display the frog and two buttons on the screen. Clicking one of the buttons should make the frog "hop" to the left. Clicking the other button should make the frog hop to the right.

We have included a picture of what the display should look like while the program is running below. In particular, in this picture we show how we would like the buttons to appear in the display.



A "starter" for this program can be found on the next two pages. Please write all the additional code needed to implement the program described on this starter.

```
public class ToTheHop extends WindowController
                        implements ActionListener
{
    // position frog is displayed initially
    private static final int FROGX = 160;
    private static final int FROGY = 10;

    // distance frog should hop
    private static final int FROGWIDTH = 70;
    // place your instance variable declarations here

    public void begin()
    {
        frog = new VisibleImage( getImage("froggy.gif"),
                                FROGX, FROGY, canvas );
    }

    // place additional method defs on the following page.
```


4. Short questions:

- a. Please rewrite the following method so that it does the same thing but uses better style (and is more compact):

```
public boolean sameNums(int x, int y) {  
    if (x == y) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

- b. Simplify the following code in which `continuing` is a boolean variable:

```
if (continuing) {  
  
} else {  
    System.out.println("Still working");  
}
```

- c. I stated in class that local variables are generally to be preferred over instance variables when writing class and method definitions. Under what circumstances must instance variables be used instead of local variables.

- d. Given the interface Alphabet below, which of the following classes could implement Alphabet?

```
public interface Alphabet {  
    public void x( Color aColor );  
    public double y( int number );  
    public boolean z( Location point );  
}
```

- i. Class: A

```
Methods: public void x( Color newColor )  
         public int y( int number )  
         public boolean z( Location point )
```

- ii. Class: B

```
Methods: public void w()  
         public void x( Color someColor )  
         public double y( int number )  
         public boolean z( Location point )
```

- iii. Class: C

```
Methods: public void w()  
         public void x( Color aColor )  
         public double y( int number )
```

- iv. Class: D

```
Methods: public void x( Color newColor )  
         public int z( int number )  
         public boolean y( Location point )
```