CMSC 113: Computer Science I
Review Questions for Exam #1

During the actual exam, you will have access to whatever printed resources you like, but no electronic resources of any sort.

Part I. Each of the following programs draws a rectangle. For each problem, say where will its upper-left corner be with respect to the app window's upper-left hand corner (i.e., global coordinates). You may assume all the programs are correct and indeed draw a rectangle, and that all the necessary import statements have been included (but are left out of the test questions).

1.
```
public class Problem1 extends GraphicsProgram
{
     @Override
     public void run()
     {
          GRect rect = new GRect(70, 80, 90, 100);
          add(rect);
     }
}
```

2.
```
public class Problem2 extends GraphicsProgram
{
     @Override
     public void run()
     {
          GRect rect = new GRect(90, 60, 10, 10);
          rect.move(20, 20);
          add(rect);
     }
}
```

3.
```
public class Problem3 extends GraphicsProgram
{
     @Override
     public void run()
     {
          GRect rect = new GRect(30, 50, 10, 10);
          rect.setLocation(70, 100);
          add(rect);
     }
}
```

4.
```
public class Problem4 extends GraphicsProgram
{
     @Override
     public void run()
     {
          GRect rect = new GRect(20, 40, 10, 10);
          GOval oval = new GOval(80, 100, 10, 10);
          rect.setLocation(oval.getX(), rect.getY());
          add(rect);
     }
}
```

5.
```
public class Problem5 extends GraphicsProgram
{
     @Override
     public void run()
     {
          Problem5Object obj = new Problem5Object();
          obj.setLocation(30, 60);
          add(obj);
     }
}

public class Problem5Object extends GCompound
{
     public Problem5Object()
     {
          GRect rect = new GRect(0, 0, 10, 10);
          add(rect);
     }
}
```

Part II. Code reading

6. Consider the following program and answer the questions below it.

```
public class ShowHouse extends GraphicsProgram
{
     private House house;
     private int numClicks;
     private GLabel clickLabel;

     @Override
     public void run()
     {
          house = new House();
          house.setLocation(100, 100);
          add(house);

          clickLabel = new GLabel("Clicks: 0", 5, 195);
          add(clickLabel);

          addMouseListeners();
     }

     @Override
     public void mousePressed(MouseEvent e)
     {
          house.setLocation(e.getX(), e.getY());
          numClicks = numClicks + 1;
          clickLabel.setLabel("Clicks: " + numClicks);
     }
}

public class House extends GCompound
{
     public House()
     {
          GRect frame = new GRect(-20, -40, 40, 40);
          add(frame);
          GLine line1 = new GLine(-30, -30, 0, -60);
          add(line1);
          GLine line2 = new GLine(0, -60, 30, -30);
          add(line2);
     }
}
```

a. What are the classes declared in this program?

b. How many fields are declared in the main program class?

c. What are their names?

d. How many fields are declared in the compound object?

e. What are their names?

f. How many methods are declared in the compound object?

g. What are their names?

h. Will this program react to the mouse? the keyboard? the passage of time?

i. Draw a picture of what this program will look like when it starts up:

Part III. What does this program do?

7. Write out what the following programs print.

   a.
```
public static void main(String[] args)
{
   int i;
   for(i = 2; i < 15; i += i - 1)
   {
        System.out.println(i);
   }
   System.out.println(i);
}
```

   b.
```
public static void main(String[] args)
{
   for(int num = 384950; num > 0; num /= 10)
   {
        System.out.println(num % 10);
   }
}
```

Part IV. Writing programs.

8. Write a console program that asks the user for a number and then prints out the number of divisors (less than the number itself) it has. So, if the user enters 12, the program would output 5, because 12 is divisible by 1, 2, 3, 4, and 6.

Along with a required console program, you will have to write two graphics programs. You will have a choice of which programs to write. Here are a few possible programs:

9. Write a graphics program simulating a car crash. Draw the car as a rectangle, moving into the applet from the left side of the screen. Draw a wall as a line going down the center of the applet. When the front end of the car hits the wall (when the right side of the rectangle touches the line), the car immediately comes to a stop and does not move again.

10. Write a graphics program to countdown to takeoff. The program starts with the text "Count: 5" displayed in the center. After every click, the count goes down. When the count reaches 0, the words "Blast off!" appear below the countdown. Further clicks do not affect the applet in any way.

11. Write an applet with a 100x100 box in the center. If the user clicks in the left half of this box, it grows bigger. If the user clicks in the right half of this box, it shrinks smaller. There is no limit to how big or small the box can get. The box must stay centered as it changes size.

12. Write a program showing a rocket taking off. Represent the rocket as an upward-pointing triangle (made of three lines – don't worry about polygons!) starting at the bottom of the applet. When you click once, the rocket starts moving up. It should *not* get faster every time you click.