ANSWER KEY
Final Exam
Computer Programming 230
Dr. St. John
Lehman College
City University of New York
Thursday, 20 May 2010

1. True or False:

   (a) T In Alice and Java, if statements can be nested inside other if statements.

   (b) F In Alice, an event only occurs as a result of user action.

   (c) T The index of an array always starts with 0.

   (d) F All variables in Alice and Java are global.

   (e) T Some methods in Alice and Java are called automatically.

   (f) F In Java, only components have the keyboard focus.

   (g) T In Java, the method *size*() returns the number of elements in an array.

   (h) F In Java, arrays cannot be parameters (inputs) to a method.

   (i) F In Java, you can only read in files, not print to files.

   (j) F In Java, all exceptions must be handled by the method that generates them.

2. Write the Java code that declares

   (a) a integer `i` that holds the number 1:
   ```
   int i = 1;
   ```

   (b) a double `tax` which is `7.75`:
   ```
   double tax = 7.75;
   ```

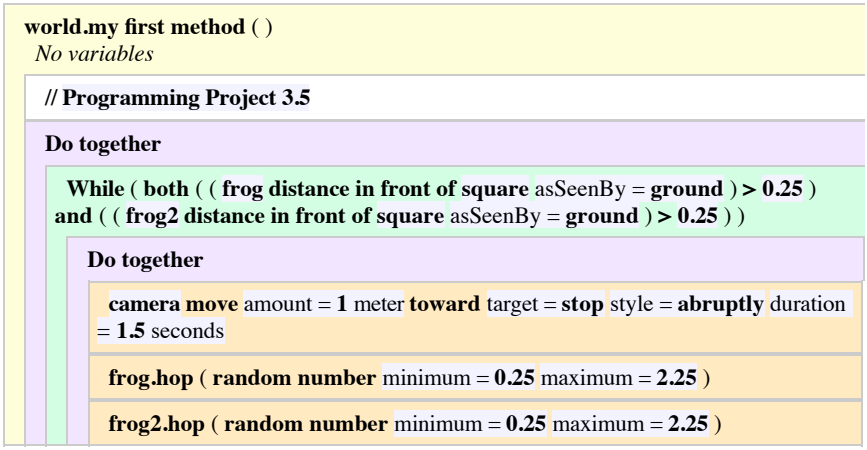
   (c) a string `myName` that holds your name:
   ```
   String myName = "Katherine";
   ```

   (d) an object `upperLeft` of the class `Point`:
   ```
   Point upperLeft;
   ```

   (e) an array `friends` of 10 `Person` objects:
   ```
   Person[] friends = new Person[10];
   ```

3. What happens when the code is run?

(a)



Two frogs each hop a random distance while each of them is more than a 1/4 meter from the square

(b)

A ball falls to the ground (while rotating forward). The balls continues to bounce, decreasing the height of the rebound by half every time. It continues the bounces until the height are less than half the height of the ball.

4. What is the output of the following code fragments:

(a)

```
int numtimes = -1;
while ( numtimes <= 0 )
```

Output:

```
Hi!Hi!Bye!
```

```
{
    System.out.print("Hi!");
    numtimes++;
}
System.out.print("Bye!");
```

(b)

Output:

```
boolean done = false;
int total = 1;
while ( !done )
{
    if ( total > 4 )
    {
        done = true;
    }
    total = total*2;
}
System.out.println(total);
```

```
8
```

(c)

Output:

```
int i, j;
for ( i = 0 ; i < 3 ; i++)
{
    for ( j = 0 ; j < i ; j++)
    {
        System.out.print("+");
    }
    System.out.println();
}
```

```
+
++
```

(d)

Output:

```
int i, j;
for ( i = 0 ; i < 6 ; i++)
{
    for ( j = 0 ; j < 3 ; j++)
    {
        if ( i%2 == 0 )
        {
            System.out.print("+");
        }
        else
        {
            System.out.print("-");
        }
    }
    System.out.println();
}
```

```
+++
---
+++
---
+++
---
```

5. What is the output?

(a)
```
if ( ( true ) && ( false ) )
    System.out.println("Yes");
```

```
else
    System.out.println("No");
```

Output: No

(b)
```
boolean tobe = true;
if ( tobe || !tobe )
    System.out.println("Yes");
else
    System.out.println("No");
```

Output: Yes

(c)
```
int x = 1, y = 2, z = 3;
if ( x+y*z <  10 )
    System.out.println("Yes");
else
    System.out.println("No");
```

Output: Yes

(d)
```
int number = -6;
boolean ispositive = ( number > 0 );
boolean ismult3 = ( number % 3 == 0 );
if ( ispositive || ismult3 )
    System.out.println("Yes");
else
    System.out.println("No");
```

Output: Yes

(e)
```
int seconds = 120;
if ( seconds%60 == 0 && seconds%3600 != 0 )
    System.out.println("Yes");
else
    System.out.println("No");
```

Output: Yes

6. Assume the following class definition:

```
public class Mystery {
    public int number;
    public String message;
    public Mystery()
    { number = 3; message = "Hello"; }
    public String toString()
    {   System.out.println(number+" "+message); }
```

```
    public void query()
    {   int i;
        System.out.print(message);
        for ( i = 0 ; i < number ; i++ )
            System.out.print("!");
        System.out.println();
    }
}
```

and the following code has been executed:

```
Mystery first = new Mystery();
Mystery second, third;
first.number = 2;
first.message = "Hi";
second = new Mystery();
second.number = 2*first.number;
third = first;
```

What is the output from the following statements?

(a) `System.out.print(first);`

Output: `2 Hi!`

(b) `first.query();`

Output: `!!`

(c) `System.out.print(second);`

Output: `4 Hello`

(d) `second.query();`

Output: `!!!!`

(e) `System.out.print(third);`

Output: `2 Hi!`

7. Examine the class below and answer the following:

   (a) How many constructors does this class have? `0`

   (b) Does the panel have an associated action listener? `no`

   (c) Does the panel have an associated mouse listener? `yes`

   (d) What does the `paintComponent()` method do, *in your own words*:
   `The paintComponent draws the last 10 dots on the screen.`

```
public class DotsPanel extends JPanel
{
   private final int SIZE = 6;  // radius of each dot
   private ArrayList<Point> pointList;
   public DotsPanel()
   {
      pointList = new ArrayList<Point>();
      addMouseListener(new DotsListener());
      setBackground(Color.black);
      setPreferredSize(new Dimension(300, 200));
   }
   public void paintComponent(Graphics page)
   {
      super.paintComponent(page);
      page.setColor(Color.green);
      for (Point spot : pointList)
      {
        if (pointList.indexOf(spot) >= pointList.size()-10)
           page.fillOval((int)spot.getX()-SIZE, (int)spot.getY()-SIZE, SIZE*2,
             SIZE*2);
      }
      page.drawString("Count: " + pointList.size(), 5, 15);
   }
   private class DotsListener extends MouseAdapter
   {
      public void mousePressed (MouseEvent event)
      {
         pointList.add(event.getPoint());
         repaint();
      }
   }
}
```

8. (a) Write a `for`-loop that prints out the numbers from -5 to 0:

```
-5 -4 -3 -2 -1 0
```

```
for (int i = -5; i <= 0; i++)
  System.out.print(i+" ");
```

   (b) Write a `while`-loop that reads characters from the `Scanner` object `line` while there are still characters on the line and prints out each character scanned on a separate line. For example, if `hi mom` is entered, you should print:

```
h
i
```

```
m
o
m


int ch = 0;
while ( ch < line.length() )
{
  char current = line.charAt(ch);
  System.out.prinln(current);
  ch++;
}
```

9. You have just been accepted a job with the a local bookstore. Your first assignment is to keep track of inventory of books at the store. Your predecessor, before quitting, began writing a `Book` class. Each of the methods of the class is proceeded by a comment that explains what the method should do. Fill in each method with the appropriate code:

```
public class Book
{
    public String title;    //The title of the book
    public int numCopies;   //Number of copies of the book
    public int numRequests; //Number of people requesting book
    public double price;    //Price of book

    public Book(String t, int n, double p) {
       title = t;  numCopies = n; price = p;
    }
    /* Prints all the information about the book: */
    public String toString()
    {

        return(t+" ("+numRequests+"/"+numCopies+") $"+price);

    }
   /* Calculates and returns the number of books available (ie the difference
       between numCopies and numRequests).  */
   public int numAvailable()
   {

        return(numCopies-numRequests);

   }
    /* Returns true if there's 1 or more books in stock, otherwise returns false*/
   public boolean inStock()
   {
```

```
        if ( numCopies > 0 )
          return(true);
        else
          return(false);

   }
}
```

10. Create a new class called `Rectangle` that extends the abstract class `BoundedShape` below. Your `Rectangle` class should have two variables to store points and a constructor that takes two points and a color as input and stores them. You should also write a method `draw()` that draws a rectangle using the information stored in the class.

```
public abstract class Shape
{  protected Color color;
   public void abstract draw(Graphics gc) { }  }
public abstract class BoundedShape extends Shape
{  protected Point upperLeft;
   protected int width, height;
   protected boolean filled;
   //  Creates and returns a point representing the upper left corner of a
   //  bounding rectangle based on two points.
   protected Point determineUpperLeft(Point p1, Point p2)
   {
      int x = (int) Math.min(p1.getX(), p2.getX());
      int y = (int) Math.min(p1.getY(), p2.getY());
      return new Point(x, y);
   }
}

//********************************************************************************
//  Rectangle.java       Programming with Alice and Java
//
//  Represents a rectangle that can be drawn in a particular graphics context.
//********************************************************************************

import java.awt.*;

public class Rectangle extends BoundedShape
{
   //-----------------------------------------------------------------------------
   //  Sets the characteristics of the rectangle based on two points.
   //-----------------------------------------------------------------------------
   public Rectangle(Point p1, Point p2, Color shade, boolean isFilled)
   {
```

```
      upperLeft = determineUpperLeft(p1, p2);
      width = (int) Math.abs(p1.getX()-p2.getX());
      height = (int) Math.abs(p1.getY()-p2.getY());
      filled = isFilled;
      color = shade;
   }

   //-----------------------------------------------------------------------
   //  Draws the rectangle in the specified graphics context.
   //-----------------------------------------------------------------------
   public void draw(Graphics gc)
   {
      gc.setColor(color);

      int x = (int) upperLeft.getX();
      int y = (int) upperLeft.getY();

      if (filled)
         gc.fillRect(x, y, width, height);
      else
         gc.drawRect(x, y, width, height);
   }
}
```