1. Write code to answer each of the following questions. For this question, it is not necessary for you to define constants or write comments.

   a. (5 points) Declare an instance variable to hold an array of filled rectangles.

   b. The JComboBox class provided by Java is used to define menus. Some of the methods defined for JComboBox are:

      public int getSelectedIndex()

         getSelectedIndex returns the numeric position of the item selected by the user.

      public void setSelectedIndex(int anIndex)

         setSelectedIndex sets the item selected in the menu to the one at the position defined by anIndex.

      public boolean isEnabled()

         isEnabled returns true if the JComboBox is currently enabled, meaning that the user can interact with it.

      Use the following variable declarations when answering the questions below.

         JComboBox box;
         int index;
         boolean enabled;

      i. (5 points) Write a statement that gets the selected index from box, assigning the result to index.

      ii. (5 points) Write a statement to change the selected item of box to the one at position index.

      iii. (5 points) Write a statement to determine if box is enabled, assigning the result to enabled.

2. (15 points) The following program presents some numbers to the user that represent the beginning of a numeric sequence. The user enters what she believes to be the next number in the JTextField. If the user's answer is correct the program displays "Right!". If it is wrong, the program displays "Wrong!".

   Below is a nearly complete version of the program. **The only thing that it is missing are the declarations of the variables and parameters:**

correctAnswer	gap	gapGenerator	guess
length	nextNumber	result	seq
sequenceEnd	sequenceStart	southPanel	startSequence
userGuess			

   Add the variable and parameter declarations being careful to choose the correct type for the variable. Use a local variable unless doing so would cause the program to either not compile or not work properly. Write your answer directly on the program below in the proper locations. To help you see where the variables are used, I have written the variable uses in boldface.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import objectdraw.*;

// Program that generates a sequence.  It shows the user all the
// numbers in the sequence except the last one.  The user guesses the
// last value and is told if he/she is correct or not.
public class SequenceGenerator extends FrameWindowController
                               implements ActionListener
{
   private static final int SEQUENCE_LENGTH = 5;
   private static final int SEQUENCE_LEFT = 10;
   private static final int SEQUENCE_TOP = 10;
```

```
   // Create the display and the sequence to guess.
   public void begin() {









      // Create the sequence
      seq = new Sequence (SEQUENCE_LENGTH);

      // Display all but the last number of the sequence
      startSequence = "Initial sequence = " + seq.getSequenceStart();
      new Text (startSequence, SEQUENCE_LEFT, SEQUENCE_TOP, canvas);
      result = new Text ("What's next?", 10, 30, canvas);

      // Put the guessing area on the display
      southPanel = new JPanel();
      southPanel.add (new JLabel ("Guess:"));
      guess = new JTextField (10);
      add (guess, BorderLayout.SOUTH);
      guess.addActionListener (this);
   }

   // When the user enters a guess, compare it to the correct answer
   // and tell the user if he/she was correct or not.
   public void actionPerformed (ActionEvent evt)
   {









      userGuess = guess.getText();
      correctAnswer = "" + seq.getSequenceEnd();
      if (userGuess.equals (correctAnswer))
      {
            result.setText ("Right!");
      }
      else
      {
         result.setText ("Wrong!");
      }
   }
}
```

```
import objectdraw.*;

/* This class creates a sequence starting at 1 with a fixed, but random
   gap between the numbers in the sequence. */
public class Sequence
{









   // Create a sequence of a given length.
   public Sequence (                    )
   {









      // Choose a gap between numbers
      gapGenerator = new RandomIntGenerator (1, 10);
      gap = gapGenerator.nextValue();

      // Fill in the numbers in the sequence
      nextNumber = 1;
      sequenceStart = "1";
      for (int i = 1; i < length - 1; i++)
      {
         nextNumber = nextNumber + gap;
         sequenceStart = sequenceStart + ", " + nextNumber;
      }
      sequenceEnd = nextNumber + gap;
   }

   public String getSequenceStart ()
   {


      return sequenceStart;
   }

   public int getSequenceEnd ()
   {


      return sequenceEnd;
   }
}
```

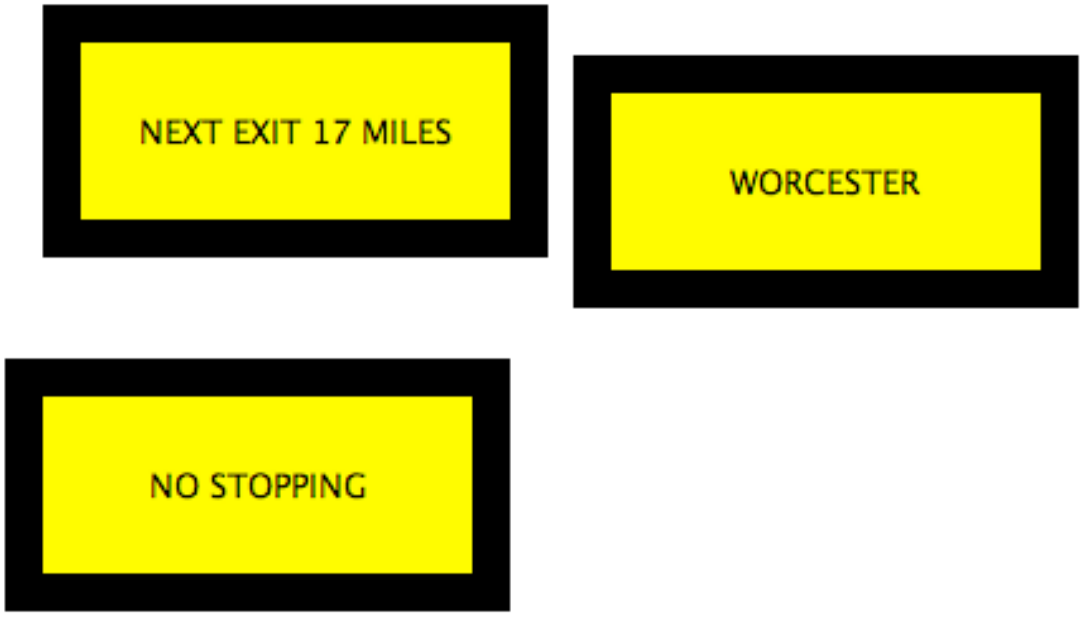
3. (20 points) As you drive along a major highway (like the Massachusetts Turnpike) you see a lot of road signs that all have a fairly common design. They are rectangular. They are composed of two colors --- one used for the background and another one used for the text that appears on the sign and as a border around the edge of the sign. On the other hand, the text each sign contains is different.

   We want you to write a simple Java class, RoadSign, that will create such signs as graphical objects in the drawing window. Your sign should have a yellow (Color.YELLOW) background. The border and text should be black (Color.BLACK). The text should be centered in the sign.

   All signs are the same size. The black rectangle is 200 pixels wide and 100 pixels high. The border is 15 pixels.

   For example, the signs shown below might have been created by the following instructions. The first parameter provides the text for the sign. The second parameter is the left edge of the sign. The third parameter is the top of the sign. The last parameter is the drawing canvas.

```
new RoadSign("NEXT EXIT 17 MILES", 40, 40, canvas);
new RoadSign("WORCESTER", 250, 60, canvas);
new RoadSign("NO STOPPING", 25, 180, canvas);
```



   As a reminder, here are the signatures of constructors and methods from objectdraw that you may need to answer this question:

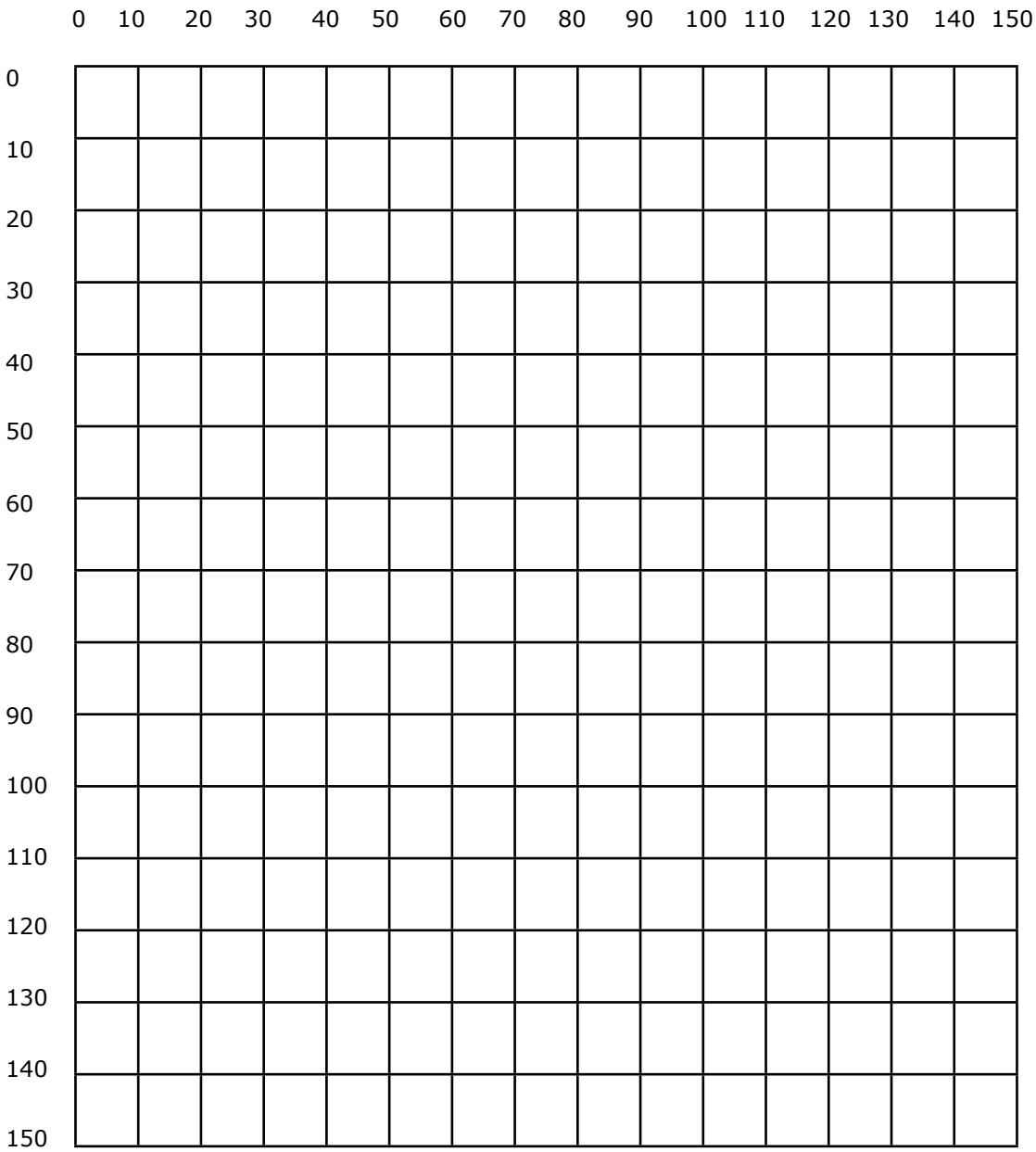
```
public FilledRect (int left, int top, int width, int height, DrawingCanvas canvas)
public FramedRect (int left, int top, int width, int height, DrawingCanvas canvas)
public Text (String words, int left, int top, DrawingCanvas canvas)
public void setColor (Color newColor)
public void move (int xoffset, int yoffset)
public void moveTo (int x, int y)
```

You should complete the constructor below to draw a sign.

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

public class RoadSign{
   // width of the black rectangle
   private static final int WIDTH = 200;

   // height of the black rectangle
   private static final int HEIGHT = 100;

   // thickness of the border
   private static final int BORDER = 15;




   public RoadSign(                                        ) {









   }
}
```

4. (15 points) The following is a simple but complete Java program that does nothing more than draw a picture on the canvas. Using the grid on the next page, draw a sketch of the picture that will be produced on the canvas provided. You may assume the canvas is large enough to hold the entire picture and that the class compiles and works correctly.

```
import objectdraw.*;

public class Mystery extends FrameWindowController {
   private static final double WHITE_LENGTH = 100;
   private static final double WHITE_WIDTH = 24;

     // Evaluates to 75
   private static final double BLACK_LENGTH = WHITE_LENGTH * .75;

     // Evaluates to 16
   private static final double BLACK_WIDTH = WHITE_WIDTH * .75;

   private static final int MAX_WHITE = 4;
   private static final int TOP = 10;
   private static final double WHITE_LEFT = 10;

   // Evaluates to 26
   private static final double BLACK_LEFT =
         WHITE_LEFT + WHITE_WIDTH - BLACK_WIDTH/2;

   public void begin () {
      int next = 1;
      double nextWhiteLeft = WHITE_LEFT;
      double nextBlackLeft = BLACK_LEFT;
      while (next < MAX_WHITE) {
         new FramedRect (nextWhiteLeft, TOP,
               WHITE_WIDTH, WHITE_LENGTH, canvas);
         new FilledRect (nextBlackLeft, TOP,
               BLACK_WIDTH, BLACK_LENGTH, canvas);
         next++;
         nextWhiteLeft = nextWhiteLeft + WHITE_WIDTH;
         nextBlackLeft = nextBlackLeft + WHITE_WIDTH;
      }
      new FramedRect (nextWhiteLeft, TOP,
               WHITE_WIDTH, WHITE_LENGTH, canvas);
   }
}
```


0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150

5. Below is a slightly modified version of the freehand drawing program (Scribble) that we saw in class. In this version, a FramedRect is first drawn on the screen. We would like this program to only allow the user to draw within the framed rectangle. Unfortunately, it doesn't quite work. When I run it, I sometimes get the red junk shown below the program.

```
1  import objectdraw.*;
2  import java.awt.*;
3
4  public class Scribble extends FrameWindowController{
5      private static final int SIZE = 200;
6      private static final Location TOP_LEFT = new Location (10, 10);
7      private Location nextLineStarts;
8      private FramedRect frame;
9
10     public void begin (){
11         frame = new FramedRect (TOP_LEFT, SIZE, SIZE, canvas);
12     }
13
14     public void onMousePress(Location point){
15         if (frame.contains (point)) {
16             nextLineStarts = point;
17         }
18     }
19
20     public void onMouseDrag(Location point){
21         if (frame.contains (point)) {
22             new Line(nextLineStarts, point, canvas);
23             nextLineStarts = point;
24         }
25     }
26 }
27
```

```
Exception in thread "AWT-EventQueue-0" java.lang.NullPointerException
    at objectdraw.Location.<init>(Location.java:58)
    at objectdraw.Line.<init>(Line.java:49)
    at Scribble.onMouseDrag(Scribble.java:22)
    at objectdraw.WindowControllerListener.mouseDragged(WindowController.java:236)
    at java.awt.Component.processMouseMotionEvent(Component.java:5602)
    at javax.swing.JComponent.processMouseMotionEvent(JComponent.java:3144)
    at java.awt.Component.processEvent(Component.java:5323)
    at java.awt.Container.processEvent(Container.java:2010)
    at java.awt.Component.dispatchEventImpl(Component.java:4021)
    at java.awt.Container.dispatchEventImpl(Container.java:2068)
    at java.awt.Component.dispatchEvent(Component.java:3869)
    at java.awt.LightweightDispatcher.retargetMouseEvent(Container.java:4256)
    at java.awt.LightweightDispatcher.processMouseEvent(Container.java:3953)
    at java.awt.LightweightDispatcher.dispatchEvent(Container.java:3866)
```

```
at java.awt.Container.dispatchEventImpl(Container.java:2054)
at java.awt.Component.dispatchEvent(Component.java:3869)
at java.awt.EventQueue.dispatchEvent(EventQueue.java:463)
at java.awt.EventDispatchThread.pumpOneEventForHierarchy(EventDispatchThread.java:269)
at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:190)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:184)
at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:176)
at java.awt.EventDispatchThread.run(EventDispatchThread.java:110)
```

a. (5 points) What user action results in the error shown above?

b. (5 points) Change the program by writing in the copy of the program below. Your revised program should draw as the original did but should not get the NullPointerException that the original program gets.

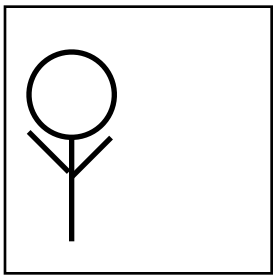
```
import objectdraw.*;
import java.awt.*;

public class Scribble extends FrameWindowController{

   private static final int SIZE = 200;

   private static final Location TOP_LEFT = new Location (10, 10);

   private Location nextLineStarts;

   private FramedRect frame;



   public void begin (){

      frame = new FramedRect (TOP_LEFT, SIZE, SIZE, canvas);

   }

   public void onMousePress(Location point){

      if (frame.contains (point)) {

         nextLineStarts = point;

      }
   }

   public void onMouseDrag(Location point){

      if (frame.contains (point)) {

         new Line(nextLineStarts, point, canvas);

         nextLineStarts = point;

      }
   }
}
```

6. In this question, I provide some image manipulation methods similar to the ones you wrote for the image manipulation lab. I would like you to tell me what the methods do. In particular, if the original picture looks as the one below, draw the picture that each of these methods produces.

   You may assume:

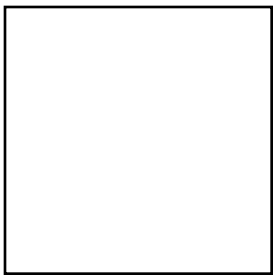
   - The methods compile and work correctly
   - The pictures are square
   - The pictures do not require scaling

   Here is the original picture:



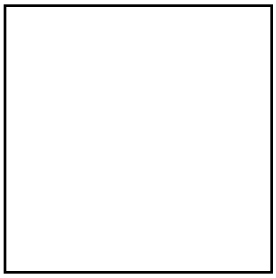
a. (5 points) Draw the picture you would see if the following method was executed when the image above is what is read from the file.

```
private void mysteryAction () {
   Picture picture = new Picture (FILENAME);
   int imgWidth = picture.getWidth();
   int imgHeight = picture.getHeight();

   Picture pic2 = new Picture (FILENAME);
   int j = 0;
   while (j < imgHeight) {
      int i = 0;
      while (i < imgWidth) {
         pic2.setPixel (j, imgWidth - i, picture.getPixel(i, j));
         i++;
      }
      j++;
   }

   pic2.createVisibleImage (0, IMAGE_TOP, canvas);
}
```



b. (5 points) Draw the picture you would see if the following method was executed when the original image shown above is what is read from the file.

```
private void mystery2Action () {
   Picture picture = new Picture (FILENAME);
   int imgWidth = picture.getWidth();
   int imgHeight = picture.getHeight();

   Picture pic2 = new Picture (FILENAME);
   int i = 0;
   while (i < imgWidth/2) {
      int j = 0;
      while (j < imgHeight/2) {
         pic2.setPixel (i*2, j*2, picture.getPixel(i, j));
         pic2.setPixel (i*2+1, j*2, picture.getPixel(i, j));
         pic2.setPixel (i*2, j*2+1, picture.getPixel(i, j));
         pic2.setPixel (i*2+1, j*2+1, picture.getPixel(i, j));
         j++;
      }
      i++;
   }

   pic2.createVisibleImage (0, IMAGE_TOP, canvas);
}
```



7. (5 points) What is the value of param when the following method returns if it is passed an array containing the numbers 3, 6, 7, 4 in that order. You can assume that the method compiles and works correctly.

```
public void mystery (int [] param) {
    int temp;
    for (int i = 1; i < param.length; i++) {
        for (int j = 1; j < param.length; j++) {
            if (param[j-1] > param[j]) {
                temp = param[j-1];
                param[j-1] = param[j];
                param[j] = temp;
            }
        }
    }
}
```