Programming Assignment 12
Due April 23

Program 1.

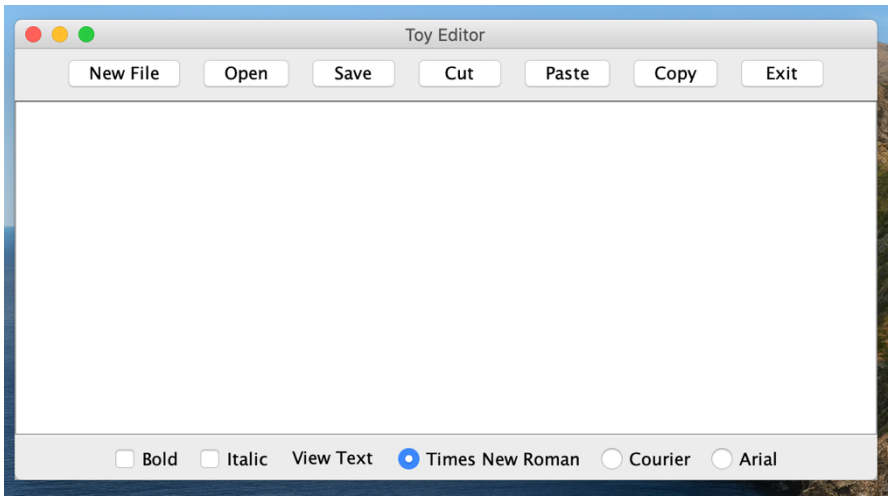
Create a very rudimentary text editor.

1. Place a text area (don't specify size) directly in a frame of size 500x500 or larger
   It will fill the entire CENTER region.
   Set the font initially to Arial, point size 12 or 14
   The text area should be placed in a JScrollPane with scroll bars as needed.
   I ft is the test area, use t.setLineWrap(true).

2. The frame should have seven buttons in the NORTH

   NewFile
   Open
   Save
   Cut
   Paste
   Copy
   Exit

At the bottom of the frame, place a panel with check boxes and radio buttons as shown below.



"View Text" is just a label—you can leave it out, if you like

3.. Once you have the GUI set up you should get the check boxes and radio buttons to work. The check boxes and radio buttons should change the font and the style in text area
To do this, **make an inner listener class that handles just check boxes and radio buttons.**

4.To make the buttons work, **make a second inner listener** just for the buttons

The cut, copy and paste buttons are easy. Assume t is the TextArea
Just call the JTextArea methods
t.cut(), t.paste() and t.copy(), where t is the text area

The NewFile, Open and Save buttons are a little trickier.

a. for NewFile…just wipe out the contents of the text area.
Remember the text area contains one long String.
Set the text in the text area to the empty string. Done!

b. for Open (opens an existing file)

Use
JOptionPane.showInputDialog(null, "Enter File Name")
to get the name of a File from a dialog box

If the filename is null or the empty string, return
Otherwise, to read a file into the text area use these statements

```
FileReader in = new FileReader(fileName);
text.read(in, null);
in.close();
```

Note the read can throw an IOException.
Use a try-catch block
If there is an error, in the catch block include the following
in the catch block:

```
JOptionPane.showMessageDialog(null,"File not Found",
"Input Error", JOptionPane.ERROR_MESSAGE);
```

c. For Save

Do something similar to Open (use an Input Dialog Box) but use

```
FileWriter out = new FileWriter(fileName);
text.write(out);
out.close();
```

d. Exit is just a call to System.exit(0).

You will be able to save files and read them back into your editor. However, text formatting (font, bold etc) is not saved.

If you build this one step at a time, you will see that it is fairly easy.

Program 2

"From chaos comes order" - Friedrich Nietzsche

. **Before coding, you should read the whole assignment.**

Note: To this lab you might use Java's Point class

The **Point** class has two ***public*** fields, x and y that represent the coordinates of a two-dimensional point (x,y

The Point constructor is

public Point(int x, int y)

So, for example, you can instantiate a point as

Point p = new Point(210,10) // or whatever coordinates you choose.

If p is a Point, because coordinates of p are public ,they can be accessed as

p.x and p.y.

You don't need getters and setters.

If you have a class called Point in your directory remove it or rename it.

Here is the problem.

Write a program which implements the following iterative algorithm.

Here is how to do it:

- Hardwire (code) into your program three points of an equilateral triangle
  p1 = (x_1, y_1),
  p2 = (x_2, y_2), and
  p3 = (x_3, y_3).
  These are screen coordinates.

Use these three points

(210,10) , (10,410) , and (410,410)

They determine an equilateral triangle.

Store these three points in an array of Point objects.
(See the red code in the main method below)

- Let *w* be one of the three vertices, **chosen at random**. Use the Random class to pick one at random. Generate 0,1, or 2.

- Repeat " forever " **//10000 drawn points is enough**
  {

  Pick one vertex p1, p2, or p3 at random.
  Call this point *v*.
  Draw a point, *mid,* exactly halfway between *w* and *v*.
  Set *w* equal to *mid*
  }

You should do the drawing on a panel that means you must override paintComponent

So, define a **class Chaos that extends JPanel** and do all the drawing on this panel.

- The data of the class should be
  an array of three Point objects (the verticies)
  as well as three other Point objects --w, v, and mid, as described in the algorithm

- A one-argument constructor should accept an array of three points.

- Set the background to white and make the points are red.

- You should override paintComponent(Graphics g) . In other words, do the drawing in the paintComponent method.

Note: Java does not provide a method that draws a single point (*x,y*). Therefore, in order to draw a point, use:

void drawRect(x, y, 1 ,1) // a 1x1 rectangle is just one pixel

of the Graphics class.

You might use the following main method:

```
public static void main(String[] args)
{
```

```
    JFrame f = new JFrame("Chaos");      // get a frame
    f.setBounds(0,0, 450,450);
   Point [] points = new Point[3]; // make an array of three points
   points [0] = new Point ( 210,10);
   points[1] = new Point ( 10,410);
   points [2] = new Point ( 410,410);
   Chaos c = new Chaos(points);       // the JPanel that you write
    f.add(c);                          // add the panel to the frame
    f.setVisible(true);
}
```

Note: the midpoint between (a,b) and (c,d) is ((a+c)/2 , (b+d)/2).

For example the midpoint between (3, 8) and (7,6) is
((3+7)/2, (8+6)/2) = (5,7)

You might include a method that computes the midpoint of a and b, such as:

```
Point getMidpoint(Point a, Point b)
{
    Point p = new Point ((a.x+b.x)/2, (a.y+b.y)/2);
    return p;
 }
```