Lab Assignment #6

Just a note…the lab assignments can reasonably take as much as 8 hours to complete at this point in the semester. Make sure you leave yourself plenty of time.

Do your own work. Your grade on exams depends heavily on how well you understand the material in the lab assignments.

Working together is not permitted. All code you submit must be your own individual work.

Programs that do not compile will receive a grade of zero.

Part A

For this part you're going to create a Java program that is both an applet and an application. Name your program Lab6A.java. Start the program with the import statements needed for JApplet, Color, Graphics, and Scanner. The class should inherit from JApplet.

Insert header comments

In the Lab6A class, write the following methods. (All 4 methods go in the same file.)

1. Write a public, static method named print5 that will accept a String and print the String on multiple lines, 5 characters per line. The last line of output may not have 5 characters.

   Example, if the String is "extemporaneous", the output would be:

   ```
   extem
   poran
   eous
   ```

2. Write a main method that will input a String from the user then call the print5 method to display the String, 5 characters per line.

3. Compile the program. When you run the program in JGrasp, it will ask if you want to run as an application or as an applet.

   Run the program and choose Run as: Application.

   Test and debug the print5 method thoroughly before going on to the next step.

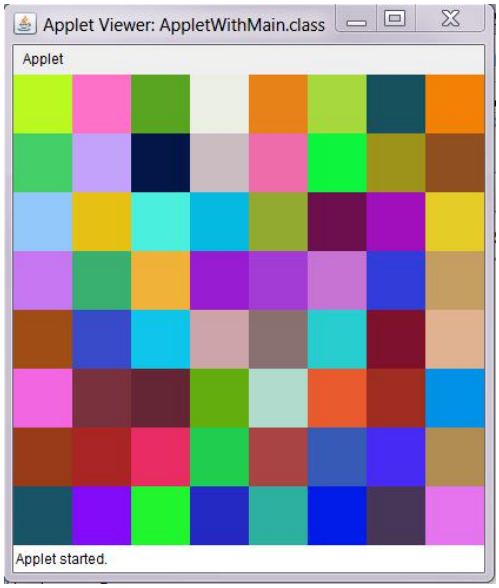
4. Write a public, static method named randomColor that creates and returns a Color object that represents a random color. Recall that a Color object can be defined by three integer values between 0 and 255, representing the contributions of red, green, and blue (its RGB value).

Reminder: generating a random integer between max and min, inclusive is done like this

```
int val = (int) (Math.random() * (max - min + 1)   +   min);
```

5. Write the paint method. (Remember that paint is not static.) In the paint method display a checkerboard pattern, 8 squares wide and 8 squares high. Each square should be 50 pixels by 50 pixels. Each square should have a random color (obtained by calling the randomColor method).

6. Compile the program. Run it as an applet this time. Test and debug the randomColor and paint methods.



Part B

1. Create a Java application. Name your program *FirstLast*.java (using your own first and last name).
2. Insert header comments
3. Implement the following pseudocode in the main method:

    a) input the name of a file from the user
    b) initialize a counter for the number of words in the file
    c) open the file (example program on page 242)
    d) initialize a counter to zero (represents the number of words in the file)
    e) loop until there are no more lines in the file

        - read a line from the file

        - Call the method named countWords (see below) to determine the number of words in the line and add that number to the counter.

    end loop

    f) output (use appropriate text to explain the output to the user)
        - the name of the file
        - the number of words in the file

4. Implement a method called countWords that accepts a String parameter, counts and returns the number of words in the String. Hint: we wrote a similar method called wordPrint in class.

```
public static void countWords (String s)
```

5. Test your program with lista.txt (see the lab assignment)
6. If your program passes the test in step 4, test your program with listb.txt.
7. If your program passes the test in step 5, design your own input file (name it mylist.txt) with at least 5 lines in it.
8. Compile, debug, repeat until it works correctly.

Submit: Lab6A.java, *FirstLast*.java and mylist.txt on Canvas.