Lab Assignment # 5

Programs must be named exactly as shown in this assignment.
Programs that do not compile will receive zero credit.

Part A: Lab evaluation of course goals #3 and #10 (10 points)

Goal #3: use a simple IDE for editing and debugging programs
Goal #10: debug and perform maintenance on code written by other programmers

Part A must be done in lab on Tuesday, Sept 30th. Skills must be demonstrated to the instructor or the TA.

The due date for parts B, C, and D is Tuesday, October 7 at 11:59 pm.

Part B: Crayon.java (8 points)

1. Design and implement a class called Crayon. The Crayon should have 2 private instance variables:

a) Color crayonColor - a Color object that represents the color to use when drawing the crayon

b) int height - an integer that represents the height of the crayon in pixels

The Crayon class should have the following methods:

c) a default constructor that initializes crayonColor to Color.red and height to 200.

d) a second constructor that accepts parameters for crayonColor and height. The header of this constructor should be:

```
public Crayon (Color c, int h)
```

This constructor should ensure that the height is a positive value. If an invalid number is passed in the h parameter, the constructor should set the height to the default value of 200.

e) an accessor "get" method for each instance variable

f) a mutator "set" method for each instance variable. The mutator for height must ensure that only positive values are stored in the height. If an invalid value is passed to a mutator, the mutator should not change the instance variable.

g) a draw method that draws a crayon. The design details are up to you but it has to be more than just a rectangle! You should draw the crayon vertical and use a width that is 1/5 of the height. Also, draw a black border around the crayon. Do not draw a paper label or logo on the crayon.

Don't remember what crayons look like?
See the image attached to this assignment.

Remember that in order to draw something on the screen, we need to know the Graphics object that is linked to the screen. This object is often called the "graphics context".

We also need to know *where* to draw so we have to pass the x and y coordinates to the draw method. In your method, the x and y values should represent the top, left corner of an imaginary rectangle that surrounds the entire crayon.

Here is the header for the draw method:

```
public void draw (Graphics screen, int x, int y)
```

2. Save and compile the class. Debug as needed before going on to the next program.

Part B: TestCrayon.java (8 points)

1. Create an applet that will act as a a test (driver) program. You should have TestCrayon.java stored in the same directory as Crayon.java, so it is not necessary to use an import statement.

a) The paint method should 8 crayons side-by-side.

b) Use a random Color for each crayon.

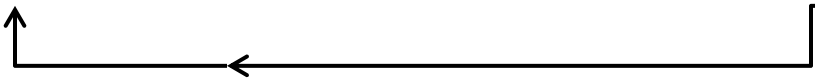
To generate a random number, use the Math.random() method - page 129.

c) Make the crayons different heights.

d) You should include statements to test every method in the Crayon class at least twice.

2. Save TestCrayon.java and compile it. If it has errors, debug as needed.

3. View your applet. Close the appletviewer. Debug. Save. Compile. Repeat.



Part C: UML diagram CrayonUML.xlsx (4 points)

Using Excel, create a UML diagram for the Crayon class. There are two examples in the textbook on page 449: the Book class and the Dictionary class.

For instance data, you must

- use the minus sign to show that the data is private
- show the name of the constant or variable
- show the type of the constant or variable

For methods, you must

- use the plus sign to show that the method is public
- show the name of the method
- show the parameter names and types
- show the return type of the method

Yes, you must use Excel. If you don't have Excel, you can go to the tutoring center at SH 155 or to one of the campus ICT labs.

Submit: Crayon.java, TestCrayon.java, and CrayonUML.xlsx on Canvas.