CS46A Exam 1

Exam Rules

- You must bring a laptop and a USB stick.
- If you want to use a power adapter, you must also bring an extension cord
- You may use any books, notes, or files on your laptop, except as noted below.
- It is a good idea to compile and run your programs if you have the time to do it.
- You may NOT browse the web, except to visit Canvas. No Google, no Piazza, no other web site.
- You may NOT communicate with anyone during the exam.
- You may NOT use the work of anyone else without attribution.
- You may NOT open any programs or web sites that allow communication with someone else. No messenger, chat, email, discussion forum, etc.
- The proctor will randomly inspect laptops. If your laptop is inspected, you need take the hands off the keyboard immediately.
- Save your solutions into Java source files (extension .java) or plain text files (extension .txt). Be sure to save your work occasionally.
- The exam is 65 minutes long. You must stop working after 65 minutes. If the proctor discovers that you worked beyond the end time, you will fail the exam.
- You have ten minutes to submit your work to Canvas or give it to the proctor (with a USB stick)
- After 75 minutes, no submissions will be accepted

1. This class has several compile-time and run-time errors. Correct as many as you find and add

   ```
   // Corrected compile-time error
   ```

   or

   ```
   // Corrected run-time error
   ```

   behind each correction that you made.

   Submit the corrected Bug1.java.

   12 points

2. Add a method

   ```
   public int getTurns()
   ```

   to this class that returns the total amount by which this bug (the solution for homework 4A) has turned. You should return an angle between 0 and 359 degrees. For example,

   ```
   Bug2 bugsy = new Bug2();
   bugsy.turn(300);
   bugsy.turn(50);
   int turns = bugsy.getTurns(); // sets turns to 350
   bugsy.turn(40);
   ```

```
turns = bugsy.getTurns(); // sets turns to 30
```

You may assume that `turn` has been called only with integers between 0 and 359.

Hints:

- You will also want to add an instance variable, and update it in another method.
- Be sure to supply javadoc for your new method.

Submit Bug2.java and Bug2Tester.java, testing the above scenario. You will get partial credit for an appropriate tester class even if your Bug2 class isn't working.

12 points

3. In Lab 4, you traced changes in the instance variables and local variables, by writing values into boxes and crossing them out. In this exercise, you will do the same tracing for a different problem. However, don't cross out old values. Simply write new values below to the old ones, and put an X when a local variable is removed. A typical run for Lab 4A would be

```
totalScore | quizCount | score | newTotalScore | newQuizCount
-----------+-----------+-------+---------------+-------------
     0     |           |       |               |
           |     0     |       |               |
           |           |   8   |               |
           |           |       |      8        |
           |           |       |               |       1
     8     |           |       |               |
           |     1     |       |               |
           |           |       |      X        |       X
           |           |   7   |               |
           |           |       |     15        |
           |           |       |               |       2
     15    |           |       |               |
           |     2     |       |               |
           |           |       |      X        |       X
```

Consider this class and the sequence of calls

```
Bug3 carol = new Bug3();
carol.moveBackward(2);
carol.turnLeft();
```

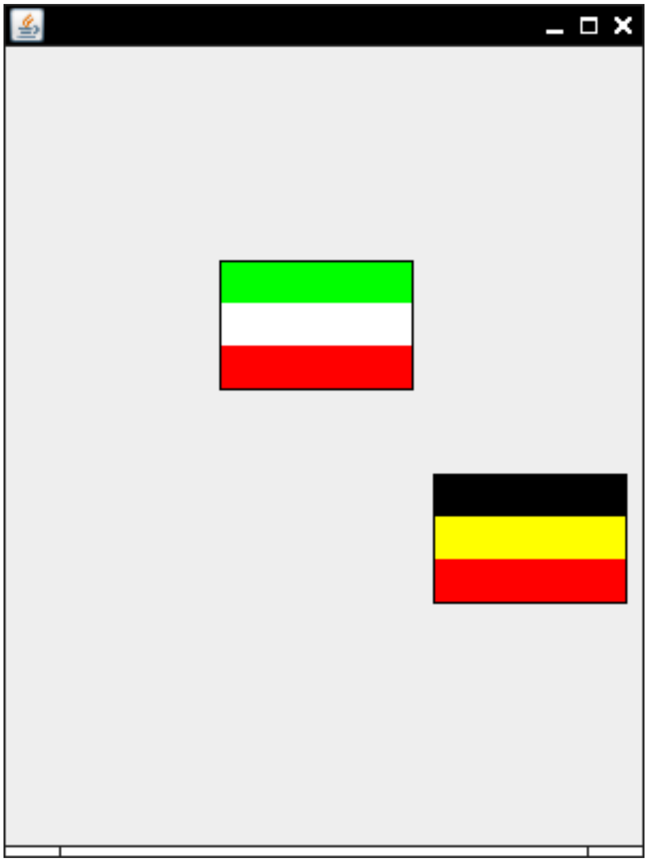
Show the values for `x`, `y`, `dx`, `dy`, `n`, and `newDx`, as they are set when executing the three statements given above. Including the removal of local variables, there are 12 changes.

Submit a file `problem3.txt` in which you fill in this form:

```
x |y |dx |dy |n |newDx
--+--+---+---+--+----
  |  |   |   |  |
  |  |   |   |  |
  |  |   |   |  |
  |  |   |   |  |
  |  |   |   |  |
  |  |   |   |  |
  |  |   |   |  |
  |  |   |   |  |
  |  |   |   |  |
  |  |   |   |  |
  |  |   |   |  |
   |   |     |     |   |
```

12 points

4. Implement a class HorizontalFlag that draws flags such as the following. A viewer and component class have been provided for your convenience. As in How To 3.2 , the height of the flag should be 2/3 of the width. As in Homework 5B, pass colors to the constructor for the top, middle, and bottom colors. Fill three rectangles, one each of the given color, and then draw a large rectangle around all three.



Submit HorizontalFlag.java.

16 points