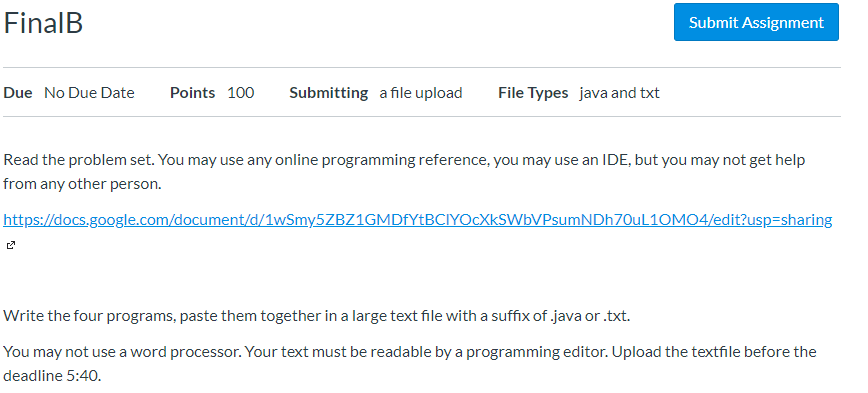
Read the problem set. You may use any online programming reference, you may use an IDE, but you may not get help from any other person.

[https://docs.google.com/document/d/1wSmy5ZBZ1GMDfYtBClYOcXkSWbVPsumNDh70uL1OMO4/edit?usp=sharing (Links to an external site.)](https://docs.google.com/document/d/1wSmy5ZBZ1GMDfYtBClYOcXkSWbVPsumNDh70uL1OMO4/edit?usp=sharing)

Write the four programs, paste them together in a large text file with a suffix of .java or .txt.

You may not use a word processor. Your text must be readable by a programming editor. Upload the textfile before the deadline 5:40.



 EE-552\_2020S Java Final B Name:\_\_\_\_\_\_\_\_\_\_\_\_\_ Score: \_\_\_\_\_\_\_  
This test is open book, open notes. You may not get help from any person or ask for help.  
Write all the code. You may use an IDE if you wish or just type it in. Please try to make it neat if you can.  
When you are done, submit a single document in the final “homework”  
You have 2.5 hours. There are 4 problems. Good luck!

1. (25 points) Write a class Statistics which reads in from a file a list of x,y points. Use Scanner to read in x,y points from a file (FileReader) and compute:
   1. The mean (average) for x and y
   2. The bounding box (minimum x, minimum y, maximum x, maximum y)
   3. Using java.util.Random,   
      <https://docs.oracle.com/javase/8/docs/api/index.html?java/util/Random.html>create a random number generator, write a method to move each point by a random uniformly distributed number from -3 to 3. Add these numbers to each point in the list.  
        
      For example, having read:  
      1.0 2.0  
      3.0 1.0  
      0.0 0.0  
      -3.0 1.0  
      4.0 0.0  
        
      The output of a. and b. should look like this:  
        
      mean: (1.0, 0.8)  
      x bounds [-3.0, 4.0]  
      y bounds [0.0, 2.0]  
        
      The following is the main you must implement:  
        
      public static void main(String[] args) {  
        Stats s = new Stats(“points.dat”); // read in the points file into the stats object  
        System.out.println(s): // print the stats as shown above  
        s.addRandomWalk(); // add uniformly distributed random numbers to every x and y  
        // print out the statistics again

2. (25 points) Write a sample window with a JButton and a JPanel using private variables like this:  
You must complete and make this code compile and run including all imports.  
Add an anonymous inner class implementing ActionListener so that when the button is pressed, it turns the JPanel p blue. Note that the method to set any component’s color is setColor(Color c);  
  
  
public class MyWindow …  
  private JButton b;  
  private JPanel p;  
….

3. (25 points) Write a class Quiz that contains an ArrayList of Questions. There are two types of questions: Multiple Choice and Fillin the blank. Multiple choice looks like:

What is 2+2?  
a) 3  
b) 5

c) Wednesday  
d) 4  
  
Fill in the blank questions look like:  
What is the title of your favorite book? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Printing a quiz should print out the name of the quiz and then print all the questions in it.  
Make the following main work:  
  
public class TestQuiz {  
  public static void main(String[] args) {  
    Quiz q = new Quiz(“Java”);  
    q.add(new MultipleChoiceQuestion(“What is your favorite programming language?”,  
 new String[] {“C++”, “Java”, “Kotlin”, “Rust”});

    q.add(new FillinQuestion(“What is the word for creating an object from a class?”, “instantiate”));  
    System.out.println(q); // print the quiz and all questions

4. (25 points) Turn the above class FillinQuestion into a Java Bean. Explain the rules of a Java Bean (very short) and implement attributes:  
  
question (a string)

answer (a string)