**Name:**

**Advanced Programming in Java**

**Lab Exercise 10.2.2024**

Reference: Lesson 20 in Blue Pelican Java

The following code will apply to problems 1 – 7:

public class TvShow

{

public String actor1 = “Don Knotts”;

public static String actor2 = “Homer Simpson”;

public static int numShows = 0;

public static int x = 59;

public int y = 1059;

public String showName;

public TvShow(String nm)

{

numShows++;

showName = nm;

}

public static int numberOfShows( )

{

return numShows;

}

public void setActor1(String act1)

{

actor1 = act1;

}

}

1. At any time after several *TvShow* objects have been instantiated, how would you find out

how many shows were instantiated? (Don’t use an object to do this.)

2. Would the code inside the *numberOfShows* method still be correct if *numberOfShows*

were non-*static*? If not, why?

3. Suppose the code inside the *numberOfShows* method is replaced with the following line:

return y;

Is this legal? If not, why?

4. Write code that will print the data member *actor2*. Do this without instantiating any

objects.

5. Is the following code legal? If not, why?

TvShow.setActor1(“Jimmy Stewart”);

6. Create an instance of *TvShow* called *chrs* (pass in the String “Cheers”) and use it to

access and print the class variable *numShows*.

7. Give the output of the following:

System.out.println(TvShow.x);

TvShow chrs = new TvShow(“Cheers”);

System.out.println(TvShow.x);

System.out.println(chrs.x);

TvShow hc = new TvShow(“History Channel”);

hc.x = 160;

System.out.println(TvShow.x);

System.out.println(hc.x);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8. Is the following a legal declaration of a class variable? If not, why?

static public char ch = ‘K’;

9. Write code that will cause the variable *zxb* to be a *static* variable. The variable *zxb* is a

*double*.

10. Write code that will cause *sn* to be a constant *static* class member. The constant *sn* is an

initially empty *String*.

11. What is the significance of the word *Math* when we use *Math.pow(3.2, 4.001)*?

12. Class variables are also called variables.

13. Assuming that the appropriate *static* import has been done, rewrite the following code

without using the class name of the *static* methods.

double xop = Math.pow(Math.sqrt(x - zv), 3.1);

14. What are the two primary reasons for using the key-word, *static*?

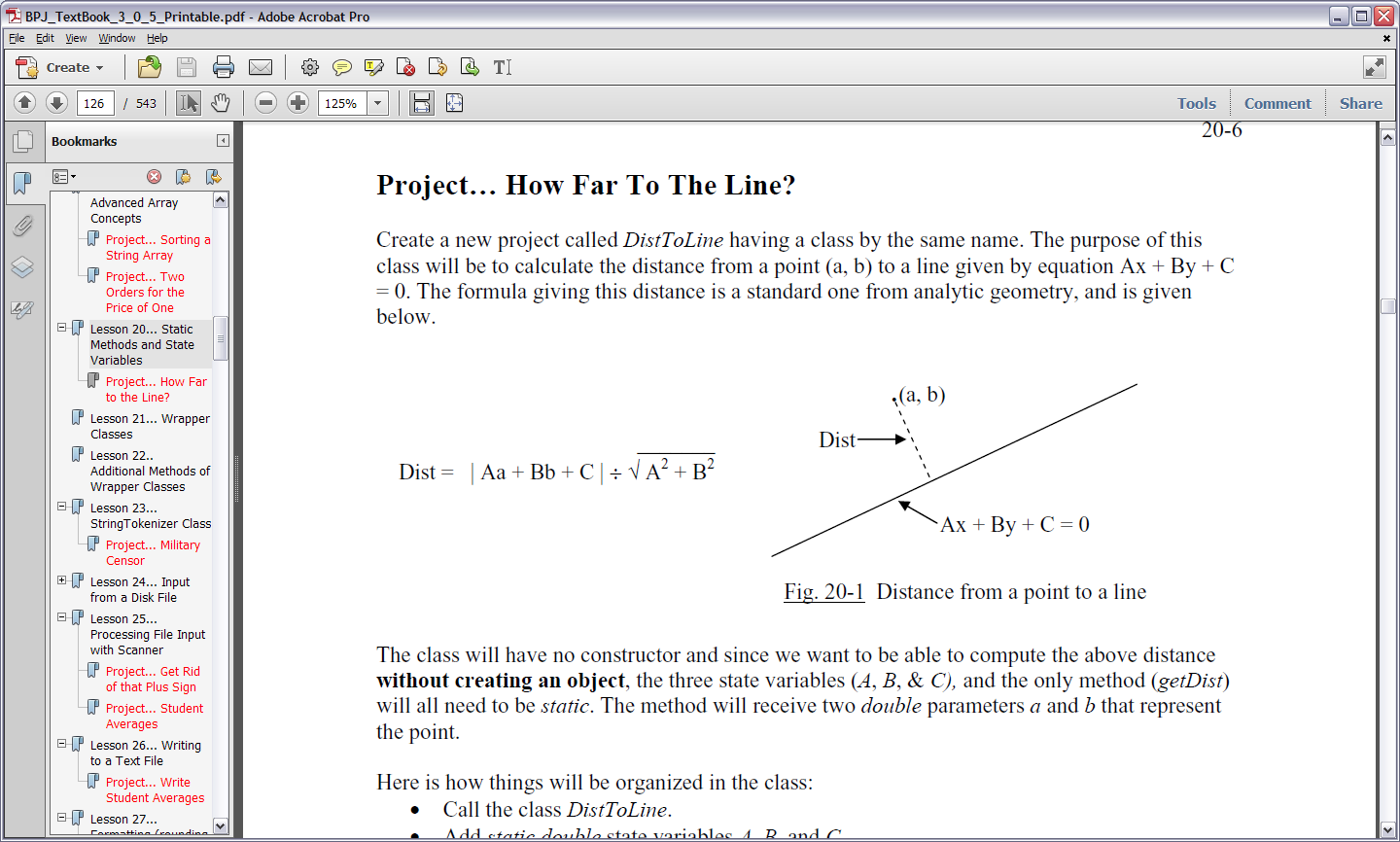
**Project… How Far To The Line?**

Create a new project called *DistToLine* having a class by the same name. The purpose of this

class will be to calculate the distance from a point (a, b) to a line given by equation Ax + By + C

= 0. The formula giving this distance is a standard one from analytic geometry, and is given

below.

The class will have no constructor and since we want to be able to compute the above distance

**without creating an object**, the three state variables (*A*, *B*, & *C),* and the only method (*getDist*)

will all need to be *static*. The method will receive two *double* parameters *a* and *b* that represent

the point.

Here is how things will be organized in the class:

* Call the class *DistToLine*.
* Add *static double* state variables *A*, *B*, and *C*.
* Create the signature of the *static* method *getDist*. It will receive doubles *a* and *b*. It will
* return a *double* representing the calculated distance.
* In the body of this method, implement the distance formula above and return that value.

Create a *Tester main* class as part of this project that will:

* Set the *static* state variables *A*, *B*, and *C* with the corresponding values of the desired line.
* Call the method *getDist* and pass as arguments the coordinates of the desired point.
* Print the returned *double* as the distance from the point to the line.

Typical output of the *Tester* class is shown below:

Enter the A value for the line: 2.45

Enter the B value for the line: 4

Enter the C value for the line: -8

Enter the x coordinate of the point: 2.17

Enter the y coordinate of the point: -4

Distance from the point to the line is: 3.9831092774319026

**When you have completed this project, submit your documented source code as well as this document with the questions answered.**