CITP 190 – Introduction to Java

Project 9

This project requires you to create one project. The project will work with a Console class you create. Please be sure to download and unzip the project start files.

To receive full credit for this project you must submit the following:

- A detailed design diagram for each method you create.
- The source code for each program (the .java file) following the coding standards.
- The bytecode for each program (the.class file)
- Proof of the correctness of your output using the test data provided. Please note that you must provide proof for all test data. You may provide additional test data.
- A capture of the output of the program. The output must show all data from your proof.
- The Javadoc files (The doc folder for the project).

Submit all files as one (1) ZIP file to the Project 9 Drop Box in the course site.

Grading:

Program design	5 points
Design diagrams	5 points
Following course standards	5 points
Proof	1 point
Screen captures	2 points
Javadoc files	2 points

Important notes:

1. Incorrect calculations will result in a 0 grade for this project.
2. Output that is not presented as shown (including spaces and spelling) will result in a 0 grade for this project.
3. Code that does not follow the standards will result in a 0 grade for this project.
4. Uploading more than one zip file will result in a 0 grade for this project.
5. Not using all the test data provided will result in a 0 grade for this project.

Project 9: Create, package, and document the Console class

Console

```
Welcome to the Console Tester application

Int Test
Enter an integer between -100 and 100:
Error! This entry is required. Try again.
Enter an integer between -100 and 100: x
Error! Invalid integer value. Try again.
Enter an integer between -100 and 100: -101
Error! Number must be greater than -101
Enter an integer between -100 and 100: 101
Error! Number must be less than 101
Enter an integer between -100 and 100: 50

Double Test
Enter any number between -100 and 100:
Error! This entry is required. Try again.
Enter any number between -100 and 100: x
Error! Invalid decimal value. Try again.
Enter any number between -100 and 100: -101
Error! Number must be greater than -101.0
Enter any number between -100 and 100: 101
Error! Number must be less than 101.0
Enter any number between -100 and 100: 50

Required String Test
Enter your email address:
Error! This entry is required. Try again.
Enter your email address: joelmurach@yahoo.com

String Choice Test
Select one (x/y):
Error! This entry is required. Try again.
Select one (x/y): q
Error! Entry must be 'x' or 'y'. Try again.
Select one (x/y): x
```

Operation

- This application prompts the user to enter a valid integer within a specified range, a valid double within a specified range, a required string, and one of two strings. If a user entry isn't valid, the application displays an appropriate error message.

Specifications

- Create a class named Console that can be used to display output to the user and get input from the user. Feel free to reuse your best code from any previous exercises or projects. At a minimum, this class should include these methods:

  ```
  // for output
  public void print(String s);
  public void println(String s);
  public void println();
  ```

  ```
  // for input
  public String getRequiredString(String prompt);
  public String getChoiceString(String prompt, String s1, String s2);
  public int getInt(String prompt);
  public int getIntWithinRange(String prompt, int min, int max);
  public double getDouble(String prompt);
  public double getDoubleWithinRange(String prompt, double min, double max);
  ```

- Use the class named ConsoleTestApp to test your class and make sure it's working correctly as shown in the console output.
- Store the Console class in a package named

  ```
  myUtils.util
  ```

  Then, add an import statement for this package to the ConsoleTestApp class.
- Add javadoc comments to the Console class. These comments should document the purpose, author, and version of the class. It should also document the function of each method, including any parameters accepted by the method and any value it returns.
- Generate the documentation for the Console class and store it in a directory named docs that is a subdirectory of the root directory for this project. If you use Eclipse to generate the documentation the default directory should be the correct directory.

Test Data

Entry
xx
-200
101
35
one
-100.88
100.10
100
xx@yy.com
C
y