ENSF 593/594 – Principles of Software Development
Fall 2019

Lab Assignment #6:
Interfaces, Copying Objects in Java, Exception Handling

Due Dates	
Post-Lab	Submit electronically on D2L **before 11:59 PM on Wednesday October 30**

This is an **individual** assignment.

The objectives of this lab are to understand the concept of:

1. Java interfaces
2. Copying object in Java
3. Exception handing in java
4. Introduction to Software Design Principles.


The following rules apply to this lab and all other lab assignments in future:

1. Before submitting your lab reports, take a moment to make sure that you are handing in all the material that is required. If you forget to hand something in, that is your fault; you can't use `I forgot' as an excuse to hand in parts of the assignment late.
2. **20% marks** will be deducted from the assignments handed in up to **24 hours** after each due date. It means if your mark is X out of Y, you will only gain 0.8 times X. There will be no credit for assignments turned in later than 24 hours after the due dates; they will be returned unmarked.


Lab Assignment 5 (35 marks)

Exercise - 1: Copying Objects in Java (5 Marks)

In this exercise, first you should download the following files from D2L:
`Colour6.java, Point6.java, Text6.java, Shape6.java, Circle6.java, Rectangle6.java, Prism6.java, and Geometry6.java.`

If you compile and run the program, you will see the program output showing some information about different shape objects. Now, open the file Geometry.java, and uncomment the lines of the code that are confined between commented lines EXERCISE_1_BEGINS and EXERCISE_1_ENDS. In this code segment the program is trying to copy objects r2, c2, and p2, into r1, c1, and p1. However, if you run the program and check the output you will find out that these objects are not copies of each other. In fact, as an example, r1 is just pointing to the same object that r2 is pointing (i.e. by changing r2, the description of r1 is also changing).

What to do: Your task in this exercise is to make any necessary changes to the code (any file as needed) to fix this problem so that objects of shape are properly copied.

What to hand in: Please submit **all the java files** including the files you have modified in a zip folder. You do NOT need to submit any Javadoc HTML file for this exercise.

Note: we are asking you to submit all the Java files so that we can run your project easier.



Lab Exercise - 2: Java interface and Exception Handling (10 Marks)

What to Do: Using java files downloaded for in-lab exercise 1:

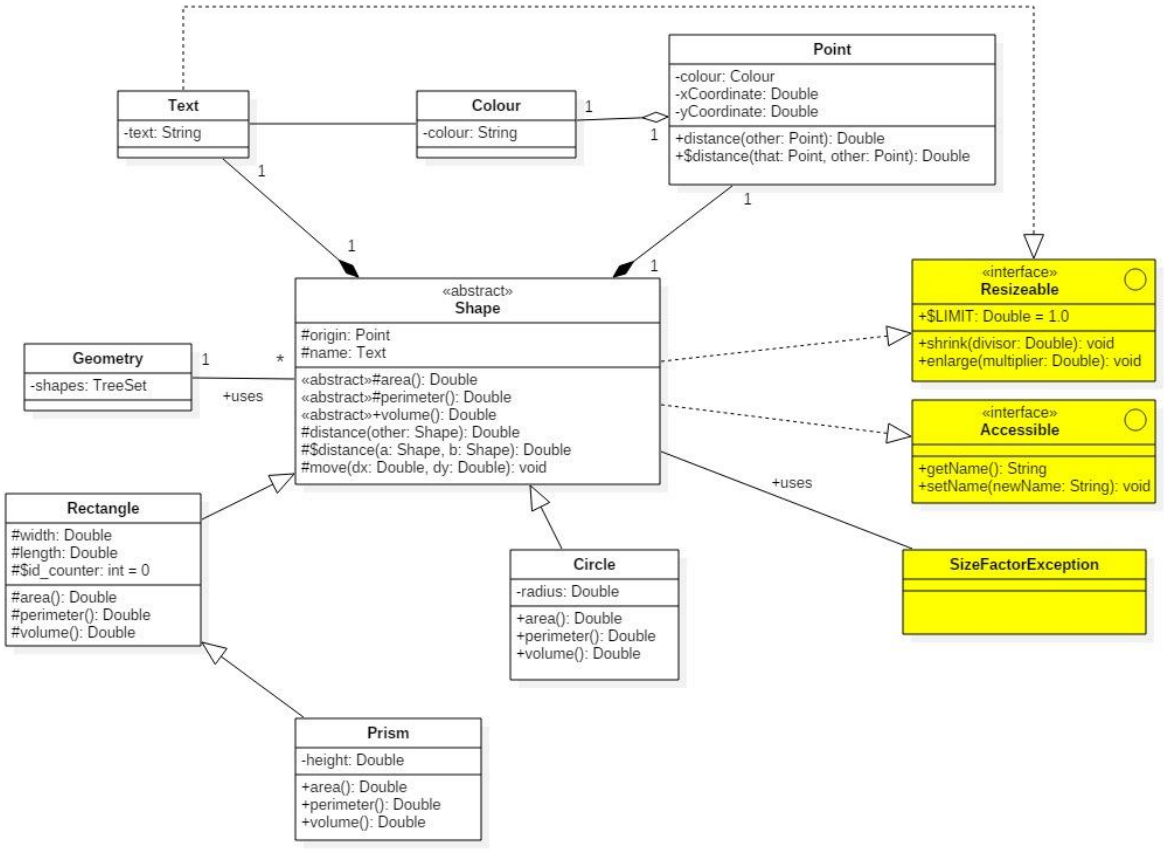
- Add two Java interfaces called `Resizable`, and `Accessible` to your program, based on the class diagram on the following page.
- Define methods `shrink` and `enlarge` in the `Resizable interface` as follows:
    - `Method shrink` is supposed to reduce the size of measurable elements of any shape, such as width, length, radius, … by a factor of n. For example, if the rectangle's width is W and its length is L this method should change them to W/n, and L/n. This method is supposed to `throw a SizeFactorException` if the value of n (i.e. the divisor) is less than `LIMIT` (e.g. 1.0) which is a variable declared in the `Resizeable` interface.
    - `Method enlarge` is supposed to enlarge the size of the measurable elements of a shape, by the factor of n. For example, the Rectangle's width W and length L should be changed to W*n, and L*n. This method is also supposed to throw a `SizeFactorException`, if the value of n (i.e. the multiplier) is less than `LIMIT` (e.g. 1.0).

- Add a class called `SizeFactorException`. Objects of this class are supposed to be thrown when the value of the arguments of methods `shrink` and `enlarge` are less that `LIMIT` (declared in the `interface Resizeable`).

- Uncomment the lines confined between commented lines `EXERCISE_2_BEGINS` and `EXERCISE_2_ENDS`, in `Geometry6.java` to test your program and to show that your code works.

What to hand in: Please submit **all the java files** including the files you have modified in a zip folder. You do NOT need to submit any Javadoc HTML file for this exercise.

Note: we are asking you to submit all the Java files so that we can run your project easier.



Lab Exercise - 3: Exception Handling (15 Marks)

For this exercise, you need to download the file `Sums.java` from D2L. This Java program allows the user to compute the sum of integers given as input. The program repeatedly asks the user if he/she wants to calculate the sum; if the answer is yes, then the program takes a sequence of integers terminated by a 0 as input, then prints their sum, and asks the same question again. If the answer is no, then the program terminates.

The program as it is written will not even pass compilation, because of uncaught (checked) exceptions. Please modify the program and add the suitable try-catch statement(s) so to cope with the following exceptions:

1. **`NumberFormatException`**: an exception of this kind means that the datum returned by `in.readLine()` does not represent an integer. If this occurs, the user should be asked to reenter the datum and continue.
2. **`IOException`**: an exception of this kind means that the `readLine()` operation could not be completed normally, for instance because of a coding error. Also in this case, the user should be asked to reenter the datum and continue.



Example of session:

```
Do you wish to calculate a sum? (y/n)
y
Please input the sequence of integers to sum, terminated by a 0
20
5
a
Invalid number. Please reenter.
2
0
The sum is 27
Do you wish to calculate another sum? (y/n)
s
Please input y or n
y
Please input the sequence of integers to sum, terminated by a 0
870
30
0
The sum is 900
Do you wish to calculate another sum? (y/n)
y
Please input the sequence of integers to sum, terminated by a 0
0
The sum is 0
Do you wish to calculate another sum? (y/n)
n
Goodbye
```

What to hand in: Please submit **all the java files** including the files you have modified in a zip folder. You do NOT need to submit any javadoc HTML file for this exercise.

Note: we are asking you to submit all the Java files so that we can run your project easier.


Lab Exercise - 4: Industry Workshop on SOLID Design Principles (5 Marks)

During the lab on **Tuesday October 29th**, we will have an industry workshop.

In this course we will continue to learn programming and software development concepts using Java. However, it is important to note that with your current level of programming skills, you should be able to transition into other programming languages.

Therefore, I have decided to have this industry workshop delivered in .NET and using C#.

.NET is very popular, especially in software development companies in Calgary and I believe it is important for you to be exposed to this technology and see the concepts you have learned using Java in .NET too.

The syntax differences between C# and Java will be mentioned in this workshop. C# is very similar to Java (C# it is widely considered to be Microsoft's version of Java).

What to do before the workshop on Thursday:

Prior to the workshop on Thursday, everyone should do the following:

1. Install and run Visual Studio Code (an IDE used for .NET development) which works both on Windows and Mac: https://code.visualstudio.com/
2. Play around with Visual Studio Code and get familiar with its environment. Create a "Hello World" application. Here is a good short tutorial: https://docs.microsoft.com/en-us/dotnet/core/tutorials/with-visual-studio-code
3. Start a project and load the files posted on D2L for exercise 4. Make sure you can run it.

What to submit:

During this workshop, you will program along with the presenter and you will continue to build the software. You will work to improve it based on the SOLID Design Principles. **Attendance is mandatory.** You will need to submit the final code.

How to submit: Include all your files for the post-lab section in one folder, zip your folder and upload it in D2L before the deadline.