**Workshop 2**

All Tasks given below must be submitted via Blackboard submission system by the deadline.

Create a zip folder which contains all the **.java** files of the tasks and screen shots for the output.

Task 1:

Design a class named **Triangle** that extends **GeometricObject**. The class contains:

* Three **double** data fields named **side1**, **side2**, and **side3** with default values **1.0** to denote three sides of the triangle.
* A no-arg constructor that creates a default triangle.
* A constructor that creates a triangle with the specified **side1**, **side2**, and **side3**.
* The accessor methods for all three data fields.
* A method named **getArea()** that returns the area of this triangle.
* A method named **getPerimeter()** that returns the perimeter of this triangle.
* A method named **toString()** that returns a string description for the triangle.

The **toString()** method is implemented as follows:

**return "Triangle: side1 = "** + side1 + **" side2 = "** + side2 + **" side3 = "** + side3;

In a triangle, the sum of any two sides is greater than the other side. The **Triangle** class must adhere to this rule. Also create the **IllegalTriangleException** class, and modify the constructor of the **Triangle** class to throw an **IllegalTriangleException** object if a triangle is created with sides that violate the rule, as follows:

/\*\* Construct a triangle with the specified sides \*/ **public** Triangle(**double** side1, **double** side2, **double** side3) **throws** IllegalTriangleException {

|  |
| --- |
| ***GeometricObject***  -color: String  -filled: boolean  -dateCreated: java.util.Date    #GeometricObject()  #GeometricObject(color: string, filled: boolean)  +getColor(): String  +setColor(color: String): void  +isFilled(): boolean  +setFilled(filled: boolean): void  +getDateCreated(): java.util.Date  +toString(): String  +getArea(): double  +getPerimeter(): double |

// Implement it

}

Write a test program that prompts the user to enter three sides of the triangle, a color, and a Boolean value to indicate whether the triangle is filled. The program should create a **Triangle** object with these sides and set the **color** and **filled** properties using the input. The program should display the area, perimeter, color, and true or false to indicate whether it is filled or not. Also display a test scenario where the exception is thrown and user will be displayed appropriate message.

Task 2:

Write a hangman game that randomly generates a word and prompts the user to guess one letter at a time, as shown in the sample run. Each letter in the word is displayed as an asterisk. When the user makes a correct guess, the actual letter is then displayed. When the user finishes a word, display the number of misses and ask the user whether to continue to play with another word. The program reads the words stored in a text file named **hangman.txt**. Words are delimited by spaces. Also make sure you properly handle exceptions in the program like “FileNotFoundException”.

|  |
| --- |
| (Guess) Enter a letter in word \*\*\*\*\*\*\* > p  (Guess) Enter a letter in word p\*\*\*\*\*\* > r  (Guess) Enter a letter in word pr\*\*r\*\* > p p is already in the word  (Guess) Enter a letter in word pr\*\*r\*\* > o  (Guess) Enter a letter in word pro\*r\*\* > g  (Guess) Enter a letter in word progr\*\* > n n is not in the word  (Guess) Enter a letter in word progr\*\* > m  (Guess) Enter a letter in word progr\*m > a  The word is program. You missed 1 time  Do you want to guess another word? Enter y or n> |

https://www.youtube.com/watch?v=VvKGhCaxHSg