

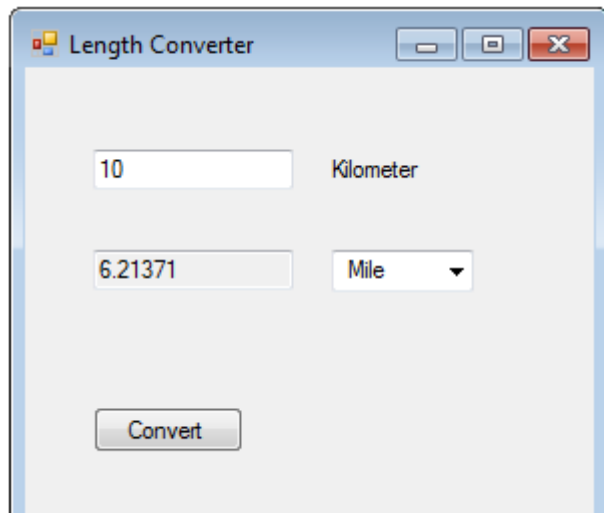
## Class Project 2

CIS 476/566 Software Architecture and Design patterns

Due: 11:59pm 03/25/2019

### Problem 1

This problem demonstrates the use of the chain of responsibility (COR) design patterns on a Length converter program (LCP) with a GUI. The LCP performs conversion from kilometer to one of the following three units: Mile, Yard, and Foot. Its user interface looks like this.



The input string specifies the amount to be converted and dropdown menu indicates which unit it will convert to. The CoR pattern will be applied to the processing of the input string to generate a number representing the converted amount. The LCP user interface is seen as a client making a request to convert the input to a given unit. Three handlers are available, one for each unit (MILE, YARD, FOOT).

The resulting processing flow for LCP is shown schematically in the next figure:



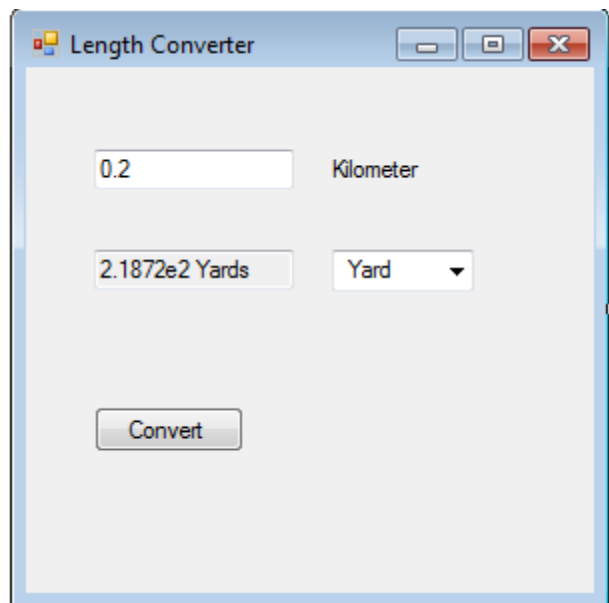
### Deliverables

- The UML class diagram.
- A SINGLE MS Word document that includes (i) screenshots showing different execution scenarios (ii) printout of the implementation code.
- The code must have graphical user interface to test the functionality.
- Softcopy of the implementation code.

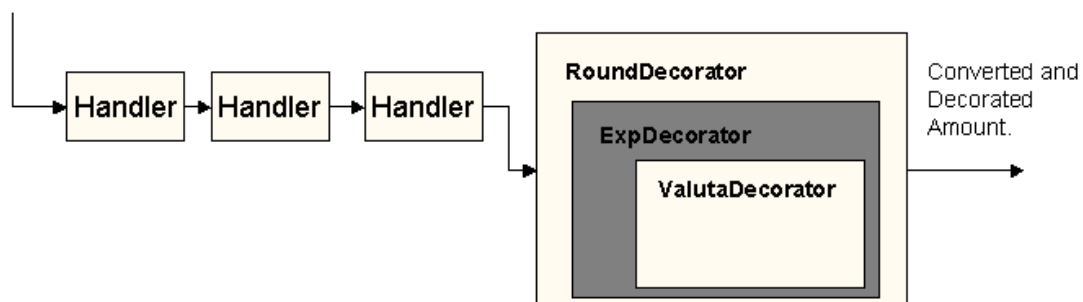
## **Problem 2**

This problem combines the decorator pattern with the CoR implemented in Problem 1. The text appearing in the output field of the LCP UI is a string that has to undergo three decorations:

- Round output to 2nd decimal (e.g., 218.723 to 218.72)
- Write output in exp. notation (e.g., 218.72 to 2.1872e2 )
- Add the unit name to the converted amount (e.g., 2.1872e2 to 2.1872e2 Yards)



The resulting processing flow (including the chain of responsibility from the previous problem) for LCP is shown schematically in the next figure:



## **Deliverables**

- The UML class diagram.
- A SINGLE MS Word document that includes (i) screenshots showing different execution scenarios (ii) printout of the implementation code.
- The code must have graphical user interface to test the functionality.
- Softcopy of the implementation code.

**Please note you should zip all your work in one .zip file and submit to project 2 in Canvas before due.**