CST 8288 Object Oriented Programming

Lab Exercise #1

Purpose Gain experience with refactoring Java code and implementing the **Strategy** pattern using:

- an **NetBeans** project,
- multiple classes organized in multiple packages,
- including javadoc & comments,
- iterative refinement of solution, and
- exporting project & source to a .zip

Activity Implement **Strategy** pattern through refactoring.

- The starter code provided implements 2 simple classes which convert Fahrenheit to Celsius and Celsius to Fahrenheit:
    - FCconverter and CFconverter
- And you are given a small main() which tests them.
- While these classes function correctly, the approach is not very generalized and difficult to extend
    - Imagine the long term goal is implement an application which is able to convert among many different types of units.
- **<u>Task #1</u>** - is to refactor the code and implement the **Strategy** pattern by:
    - creating a class called UnitConverter,
    - which includes methods to set/change:
        - type of units (starter code has one type so far, i.e. Temperature)
        - conversion behaviours (i.e. Fahrenheit to Celsius and Celsius to Fahrenheit)
- **<u>Task #2</u>** – continue to implement the **Strategy** pattern by:
    - Adding an additional type of unit and a corresponding pair of conversion behaviours.
        - The choice is up to the student but try to be different from others.
        - Conversion factors must be correct and accurate.
        - Examples: distance (kilometres vs miles), volume (liters vs Imperial gallons), weight (kilograms vs pounds), area (hectares vs acres)
- Hint: The **Strategy** example from Week2

Coding requirements:

- You may change the main() method, but the output must be similar.
- You may make small changes to the classes FCconverter and CFconverter (e.g. implement an interface)
- The "starter" code provided uses multiple classes and multiple packages. And your solution must also use multiple classes and packages.
  - e.g. add UnitConverter (which implements the Strategy pattern) to pkgUnitConverter
- the main() method should perform all I/O
  - **There is no end-user input;** there should only be output
- Be sure to include Javadoc comments for all public methods and constructors (for full marks)
- export your **NetBeans** project as a .zip
  - name your file Lastname.Firstname.Lab1.zip
  - name your project Lastname.Firstname.Lab1
  - substitute your names (of course).
- Each file must include the following comment block (with your information) :

```
/*
Student Name:
Student Number:
Course & Section #: 22S_CST8288_xxx
Declaration:
This is my own original work and is free from Plagiarism.
*/
```