CSC 207 Software Design
Winter 2015 — Exercise 3

1 Logistics

- **Due date:** ~~9:00pm Tuesday 17 March 2015~~ 9:00pm Thursday 19 March 2015
- **Group size:** Individual
- **Topics:** Design Patterns

For the rules and procedures for the exercises, including how to submit, please see the Exercises page of the course website.

2 What to do for this exercise

1. Your individual `svn` repository now contains a new directory called `E3`. It contains the starter code for this exercise. Checkout and study the starter code.
2. Complete/implement Java classes `Product`, `Shopper`, and `PriceWatchWebsite` that obey the specifications below and the descriptions in the starter code.
3. When notifying observers of a price change, use the provided class `PriceChange`.
4. To submit your work, add and commit your changes to your repository.

   Do **not** commit the files and directories generated by Eclipse, such as `bin`, `doc`, `.project`, etc. Marks will be deducted if you submit these.

3 Specifications for `Product`, `Shopper`, and `PriceWatchWebsite`

Your task is to implement `Product`, `Shopper`, and `PriceWatchWebsite` classes.

3.1 Class `Product`

A `Product` has a name, a price, and a store. Class `Product` is an `Observable` and has these methods:

- A constructor.
- Getters for name (a `String`), price (a `double`), and store (a `String`).
- A method `changePrice` that changes the price of a `Product` and notifies all its observers.
- A method `toString` that returns a `String` of the form:

  ```
  The price of PRODUCT at STORE is PRICE.
  ```

  where `PRODUCT`, `PRICE`, and `STORE` are the name, price, and store of the product. The price should be formatted to 2 decimal places. Here is an example of formatting a string in Java (see documentation for `String.format`):

  ```
  String priceFormatted = String.format("%s costs %.2f", name, price);
  ```

3.2 Class `Shopper`

A `Shopper` has a name. Class `Shopper` is an `Observer` and has these methods:

- A constructor.
- A getter for name.
- An `update` method that prints a message when a `Product` that the `Shopper` is observing changes. The message is of the form:

  ```
  SHOPPER was notified about a price change of PRODUCT at STORE to PRICE.
  ```

  where `SHOPPER`, `PRODUCT`, `STORE`, and `PRICE` are the shopper's name, product's name, store, and new price (to 2 decimal places).

3.3 Class `PriceWatchWebsite`

A `PriceWatchWebsite` has a URL. Class `PriceWatchWebsite` is an `Observer` **and** an `Observable`, and has these methods:

- A constructor.
- A getter for URL.
- A method `update` that prints a message when an object that the website is observing changes. The message is of the form:

  ```
  You are subscribed to URL.
  It was notified about a price change of PRODUCT at STORE to PRICE.
  ```

  where `URL`, `PRODUCT`, `STORE`, and `PRICE` are the website's url, product's name, store, and new price (to 2 decimal places) respectively.

  Note: this method also notifies its observers of the change.

4 Marking

The mark for correctly named files that compile and run, producing the correct output, is 3 marks. If you submit files with the correct names, but they are not in the correct directory, or do not compile, or do not belong to the correct package, or do not run, or do not produce the correct output, the solution will receive 0 marks.

5 Checklist

Have you...

- tested your code on the lab computers using **Java 1.7**?
- committed the correct files in the correct directory?
- verified that your changes were committed using `svn list` and `svn status`?
- checked the pre-marking results, made any necessary changes, and re-committed if necessary?