Sonoma State University
Computer Science Department
CS 460 – Fall 2023 – Watts

Exercise 5
(Date due: 11 December 2023; 11:59 pm)

For this exercise you will be completing, testing, and documenting the money class originally created for Exercise 2.

Specifications

1. The folder ~tiawatts/cs460pickup/Exercise5 contains:
    ```
    E5-1.cpp  makefile  moneyE5.cpp  moneyE5.h  README.txt
    ```
   The files moneyE5.h and moneyE5.cpp contain a minimal implementation of the money class. You may rename and complete these files or (better yet!) you can use your Exercise 2 files.

2. Add the prototype and implementation for a new friend method to the class:
    ```
    friend money operator * (const double & F, const money & M);
    ```
   This function will allow commutative multiplication using a money object and a numeric factor. Currently `moneyObj * value` is valid. By adding this new method, `value * moneyObj` will also be valid.

3. The makefile provided in the folder Exercise5 uses a command line variable so that you can compile and execute a variety of test programs for your money class. A test program called E5-1.cpp is included in the Exercise5 folder. To compile this program, enter
    ```
    make pgm=E5-1
    ```
   E5-1 will replace the $(pgm) references in the makefile. If there are no lexical, syntactical, or semantic errors in your money.h and money.cpp files, an executable called Ex5.exe will be generated. The output for this test program should be $0.00.

4. Make sure that you are using Doxygen style documentation. An exercise introducing Doxygen style commenting and website generation can be found at:
            http://watts.cs.sonoma.edu/cs460f23/Exercise8.pdf
   You do not need to submit this exercise.

5. Using the instructions on the following page, generate a Doxygen website for your money class.

Date due: 11 December 2023; 11:59 pm

To turn in: Create a folder called *lastname*E5 to contain your money.h, money.cpp, and README.txt files. Tar and zip your folder to a file called *lastname*E5.tgz and copy it to the course dropbox.

Creating a Doxygen website for your money class.

1. Create a folder called "CS460Exercise5" in your public_html folder and use the Linux command:

```
chmod 755 CS460Exercise5
```

to enable viewing of your documentation website.

2. Copy your money.h, money.cpp, and favorite C++ money class testing program to your new folder. Make your new folder your present working directory and enter the Linux command:

```
doxygen -g
```

to generate a Doxygen configuration file.

3. In the generated file called Doxyfile, modify the following configuration parameters:

```
PROJECT_NAME            = "CS 460 Exercise 5 – First Last"
```

Where First and Last are your first and last name

```
JAVADOC_BANNER          = YES
EXTRACT_PRIVATE         = YES
```

4. Enter the Linux command:

```
doxygen Doxyfile
```

to generate a Dyoxygen website.

5. Enter the following Linux commands to allow you to view the website:

```
chmod 755 html
chmod 644 html/*
chmod 755 html/search
chmod 644 html/search/*
```

6. Using a browser of your choice, view the generated website. The URLshould be:

```
https://blue.cs.sonoma.edu/~cslogin/CS460Exercise5/html
```

where *cslogin* is your login id for the blue is your submission id for this class.
Explore this web site – you should see entries for some, but not all, of the files in this folder.

7. To improve this website, make sure that your files are using Doxygen's tags and preferred format. Information about Doxygen tags can be found at:

```
https://www.doxygen.nl/manual/commands.html
```

The documentation at the beginning of money.h should be

```
/** ***********************************************************************
* @brief Exercise 5 – class money
* @file money.h
* @author
* @date Fall 2023
*
* @remarks This file contains the description for a C++ class called "money"
**************************************************************************/
```

8. Make sure that you are using Doxygen style comments for each of the attributes and methods in your money class. For example, the comment for private variable 'size' should be:

```
/**
* @var int size
* @brief integer variable a contains the number of unsigned values
* in th the dynamic currency array.
**/
```