CS 215 - Fundamentals of Programming II
Fall 2019 - Homework 3
10 points

Out: September 4, 2019
Due: September 9, 2019 (Monday)

Problem Statement

Write a program that will read a series of positive integers from a file, each representing a number of cents. For each integer, the program will compute the minimum number dollars, quarters, dimes, and pennies that would be needed to give out the amount in change. (Do not use nickels; they are not interesting since there would be at most one.) These results are to be output to a file in tabular format as shown below with each field having a width of 10.

The program must be implemented using a function `compute_change` that ***receives*** a positive integer representing a number of cents and ***passes back*** the number of dollars, quarters, dimes, and pennies. (I.e., there are 5 parameters to this function, and all the "results" are passed back in one call.) The passed back parameters must be implemented using C++ reference parameters. **Programs not using an appropriate function or using C reference parameters will not receive any credit.** The function prototype must be written above the main program, and the function definition must be written below the main program.

The program must accept the input and output file names as **command-line arguments**. Proper error checking of the number of command-line arguments and successful file opens must be made.

Coding Notes

This assignment is being distributed by the GitKeeper submission system. You should have received an email with the URL to clone. A reminder of the commands needed to set up this assignment:

```
$ cd <path to your class directory>
$ git clone <URL given in the assignment email>
```

This will create an working directory called `hwk03-coins` containing example input file `input.dat`. Change to this directory and create `change.cpp`.

```
$ cd hwk03-coins
$ emacs change.cpp &
```

Be sure to commit your changes early and often. **The first commit for this assignment is expected to be by Friday, September 6.** A reminder of the commands needed to do this (run in the working directory):

```
$ git add .
$ git commit -m "<description of what was completed>"
```

Here are some hints that may be helpful.

- Input should done using `>>` and an integer variable.
- The change should be computed from dollars to pennies, i.e. largest to smallest.
- The change for each denomination can be computed using integer divide and remainder (`%`).
- Value parameters are copies of the actual argument, so they can be used like local variables.
- Table column headings also can be aligned using setw( ) manipulators.

For example, the provided input file is named `input.dat` and contains the following data:

```
186
247
63
```

The program would be compiled and run using:

```
$ clang++ change.cpp -o change -Wall -std=c++11
$ ./change input.dat output.dat
```

The output file named `output.dat` should contain the following results:

```
    Amount   Dollars  Quarters     Dimes   Pennies

       186         1         3         1         1
       247         2         1         2         2
        63         0         2         1         3
```

How to submit

The program file must have a comment at the top of the file with your name and assignment number. Please note that the automated submission system requires all files be named exactly as specified in an assignment (`change.cpp`, in this case), and also requires that the output of the program be exactly as expected including whitespace, capitalization, and spelling.

Assignments are submitted to GitKeeper by pushing committed changes.

```
   $ git push
```

Only a clean repository (one where all changes have been committed) can be pushed. If a repository is not clean, git will say everything is up to date, and refuse to do the push. To see which files have been changed, but not committed, use command:

```
   $ git status
```

You can also see all the log messages by using command:

```
   $ git log
```

Note that the first commit was done by the GitKeeper system.