Computing and Algorithms I

Project 1

CS-101 **Summer, 2013**

Project 1 will have you input integer values for quarters, dimes, nickels, and pennies and output the dollar amount of the input change in the form $9,999.99 where of course the '9' digits will be replaced with the correct digits for the result. There are required comments in the project. The comments at the beginning of the code will identify the purpose of the program, you as the programmer, and your organization, that is, CS101 and section number.

In this project you will also do a design which has two parts: a data table and an algorithm. These designs will be made part of your program as comments before the main method. You will solve this problem using only one method, and that is main.

Data Table

A data table is a table that has two columns. The left column will be labeled "Variable or Constant" and the right column will be labeled "Purpose". In each row you will list a variable or a constant used in your program in the left column, and in the right column you will explain its use. In your comments you will have a section whose first line will have the words "Data Table" and the remaining lines of the section will be the data table itself.

Algorithm

An algorithm is a step by step solution of a problem which can be translated on a line by line basis into a computer program. We will write algorithms in pseudocode. Pseudocode has no absolute syntax rules, so the technical aspects of coding are ignored (such as blocks are contained within braces, ({}), and statements end in semicolons). For the purpose of this program, each line of pseudocode will correspond to one statement inside the main method. You will begin the algorithm section with a line consisting of the word "Algorithm". The next non blank line will consist of the name of the method and its arguments, in this case it will be "main(args)". The rest of the algorithm will consist of lines of pseudocode corresponding 1 to 1 with the executable statements in main. In particular, a declaration statement such as "int num" is not executable. An initialization statement such as "int num = 1" is executable and will have pseudocode something like

```
''num <-- 1''
```

.

Input

You will use JOptionPane methods to input integer values for quarters, dimes, nickels, and pennies. If the user enters values which are not integers, your program is allowed to crash. The JOptionPane message needs to be easily understood, such as "Enter the number of dimes."

Processing

The input from using JOptionPane is of course a String. The values must be converted to numeric types. At the end you need a number which you can use to print the amount of money in the desired form.

Output

Show your result using JOptionPane with a good label, such as, "Your change has monetary value:" followed by the amount of money in the proper format.

Style

Use white space (indentation, blank lines) to show the program structure. Use good variable names. Follow conventions in naming identifiers. Use constants for any constant value used in your program.

Deliverables

On Thursday August 1 at the beginning of class you will turn in a document printed from a printer containing the Java source code (of course with all comments as described above). Your Java code will be printed in portrait, not landscape, mode. None of your lines in the file are allowed to extend past the right edge of the paper. If you have multiple pages of code, they will be stapled into one package before the beginning of class.

You will submit your code (just the .java file) using Blackboard before class begins. The submission will be in the project page for this project. Projects not turned in via Blackboard and/or not turned in as a document will receive no credit.

Grading

The program itself will be graded on 10 points. If the program does not compile, it is worth 0. If the program compiles but does not solve the problem, the score will be at most 5 points, depending on how close the solution is to being correct. If the program works correctly, that is worth 8 points, with the other two points reserved for the messages being clear for the input and output windows.

The data table is worth 5 points. Full credit will be given to complete and well formatted tables.

The algorithm is worth 5 points. Full credit will be given to complete and well written pseudocode.

For a program that compiles and runs, the style score will be from 0 (hard to read) to 5 (follows all conventions and style guides) points.

The total maximum score for the project is thus 25 points.

A program which does not compile/run will be worth at most 5 points.

A program which is late (not complete and ready to hand in and demonstrate at the beginning of class on Thursday) will have a late penalty. If the program is turned in late on Thursday, the

penalty will be 5 points, on Friday the penalty will be 10 points, and on Monday the penalty will be 15 points. No program will be accepted after Monday August 5.