

Computing and Algorithms I

Project 2

CS-101

Summer, 2013

In project 2 you will design and write a program to make change. You will use the Scanner class for input and use System.out for output. The design portion of this program is the same as in project 1, and has been repeated here for your convenience.

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 statements in main.

Input

You will use an object of the Scanner class to perform input from the keyboard. You will read the amount of the purchase with two integer reads, the first read will get the dollar amount of the purchase; the second read will get the cents amount of the purchase. For example, if the purchase amount is \$5,423.71 you will read the integer value 5423 for the dollar amount, then 71 for the cents amount.

After reading the purchase price, you will read the amount tendered in the following order (all reads will be integer): the number of \$20 bills; the number of \$10 bills; the number of \$5 bills; the number of \$1 bills; the number of quarters; the number of dimes; the number of nickels; and finally, the number of pennies. There can be any nonnegative number of any of these denominations.

Processing

You will calculate the value of the amount tendered, and use it in determining how to make change. If the amount tendered was too small, your output will be a message stating that the amount tendered was not enough to cover the purchase, and will display how much more money is needed. If the amount tendered is enough to cover the purchase, your program will display the change returned in the order of \$20 bills down to pennies. Any denomination of bill or coin whose value is 0 in the change will not be displayed. For example, if the change is 12 cents, your output will display 1 dime and 2 pennies. For one dollar thirty cents, your output will display 1 \$1 bill, 1 quarter, 1 nickel. You will not use double or float for this assignment.

Your change will be the smallest number of bills and coins possible.

Output

You will be using System.out for output. The first information output should be a title, such as Change Making Program, followed by your name and section number.

Your program will then input values as described above, then write the purchase price in format corresponding to the format \$#,##0.00 applied to a double (you may assume that the largest purchase price is \$999,999.99 for this program). Label this and all other output values well. Note that you will not actually use the format just described because all of your work will be with integers. Do not convert any value to type double or float for this program.

After printing the purchase price, your program will print the amount tendered in the same format as purchase price with the same assumption for largest amount. Note that the coins tendered can amount to more than 99 cents. Test your program with coin amounts tendered such as 5 quarters, 11 dimes, 21 nickels, 101 pennies, and larger numbers.

Following the above, your program will print the change with each bill or coin denomination on its own line of output.

If the amount tendered is exactly the purchase price, your program will print a message stating that there is no change to be returned.

Style

Use white space (indentation, blank lines) to show the program structure. Note that indentation is important to show the structure of if and if else statements. Use good variable names. Follow conventions in naming identifiers. Use constants for any constant value used in your program.

Deliverables

On Thursday August 8 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.

You will turn in a paper showing sample runs of your program: one with exact change for the purchase, one with too little money for the purchase, one showing change, having some but not all of the coin amounts used in the change.

You will submit your code (just the zipped .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 clear output labels.

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.

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

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