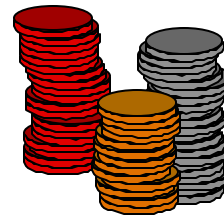


# LAB EXERCISE

## CHANGE



### Background:

Some cash register systems use automatic change machines where the coinage is automatically dispensed. This lab will investigate the problem solving and programming behind such machinery. You should use integer mathematics to solve this problem.

You will need to extract the amount of cents from dollar amounts expressed in real numbers. This will require using the type cast operator and dealing with the approximate nature of real number storage. Here is an important example:

```
double purchaseAmount, cashPaid, temp;  
int change;
```

... data input stuff

Example Values (as done in a math class):

```
temp = cashPaid - purchaseAmount;  
temp = temp - (int) (temp);  
change = (int) (temp * 100);
```

```
8.06 = 30.00 - 21.94  
0.06 = 8.06 - 8  
6 = (int) (0.06 * 100)
```

However, when the above example was run on a computer, the answer of 5 was given. Because real numbers are stored as approximations, the value of *change* comes out incorrect. You must investigate why and solve the problem.

You should solve this part of the problem **FIRST**, before you go on and calculate the correct number of quarters, dimes, nickels, and pennies.

### Assignment:

1. Write a program that does the following:

a. Generate with the RandNumGenerator (from the MBS CS)

Amount of purchase:

Where the dollars part is a random integer from 1 to 10

Where the cents part is a random real number from 0.0 to 0.99

b. Prompts the user for the following information.

Amount of cash tendered (read as a real number)

c. Calculates and prints out the coinage necessary to make correct change. You should also echo the original data entered back to the user. You should also report the number of whole dollars needed for change as well (assume that there will always be some coins returned).

2. Sample run output:

**INPUT:**

```
Amount of purchase:    23.06
Cash tendered:         30.00
```

**OUTPUT:**

```
Amount of Purchase:    23.06
Cash Tendered:         30.00

Dollars returned:      $6
Coins returned:        94¢ = 3 quarters
                        1 dime
                        1 nickel
                        4 pennies
```

3. You should format the output in some nice manner (use the class(es) provided to you by our authors).
4. You should correctly print singular versus plural endings (i.e., quarter/quarters). Use and “if” statement – this is the ONLY time you are allowed to use an if statement in this lab.
5. You must print the **\$** before the dollar amount and the **¢** after the cents! (Good luck finding the ¢ on the keyboard! You **may not** use the character map and copy/paste the character into your code; you should use what you learned in class and find a programming solution for getting that character to print out.)
6. You must use **good variable names with meaning**, constants if needed, **indentation, logic**, use of **blank lines to set off different sections of code** to ease the reading of it, **etc.**

"Introduction to Computer Science" by Cary Matsuoka, published by Institute of Computer Technology, [www.ict.org/apcs.html](http://www.ict.org/apcs.html) (modified by Dave Wittry)