Kyle Carney

CSC 1051-1

Project 2

Project 2 was one that I really benefited from. I was stuck on the math logic of how to figure out how to find the values for nickels and pennies. I think it helped that I started by getting the program functional, and then trying to figure out the algorithm. I also coded each line one at a time and ran it to make sure it worked. There were times that my code was not right, but it still produced the right answers for quarters, dimes, nickels, and pennies. That was a good lesson on why testing is important.

Now that I am looking at any errors and the rubric I see that we need descriptive variables. Initially I had the variables cents, quarters, dimes, nickels, and pennies. I then changed the variables to c, q, d, n, and p to make it easier to code. It makes sense to use descriptive words, so I went back and changed it. I always get nervous changing a lot of code that is functioning, but I tested it again 5 times, and it looks good. I enjoyed this project, because it was challenging, and I learned a lot.

1 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 2 // **CashRegisterAlgorithm.txt** Author: Kyle Carney  
 3 //  
 4 // CSC 1051 Project 2: Cash Register \*\*\*   
 5 //  
 6 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 7   
 8 variables: cents, quarters, dimes, nickels, pennies  
 9   
10 algorithm:  
11   
12 1. input cents   
13 2. print cents  
14 3. print quarters, dimes, nickels, and pennies  
15 q = c / 25  
16 d = (c - (q \* 25)) / 10  
17 n = (c - (d \* 10 + q \* 25)) / 5  
18 p = (c - ((q \* 25) + (d \* 10) + (n \* 5)) / 1  
19 if p = 1 use "penny"  
20 else use "pennies"  
21

1 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 2 // **CashRegisterAlgorithm.java** Author: Kyle Carney  
 3 //  
 4 // CSC 1051 Project 2: Cash Register \*\*\*   
 5 //  
 6 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
 7   
 8 import java.util.Scanner;  
 9 //class:  
10 public class CashRegisterAlgorithm  
11 {  
12 //method:  
13 public static void main (String[] args)  
14 {  
15   
16 //variables: cents, quarters, dimes, nickels, pennies  
17 int cents, quarters, dimes, nickels, pennies;  
18   
19 //algorithm  
20 //1. input cents  
21 Scanner scan = new Scanner(System.in);  
22 System.out.print ("Enter an amount (in cents): ");  
23 cents = scan.nextInt();   
24 //2. print cents  
25 System.out.println (cents + " cents");  
26 //3. output quarters, dimes, nickels, pennies  
27 quarters = cents / 25;  
28 dimes = (cents - (quarters \* 25))/10;   
29 nickels = (cents - (dimes \* 10 + quarters \* 25)) / 5;  
30 pennies = (cents -((quarters \* 25)+(dimes \* 10)+(nickels \* 5))/1);  
31 //4. print q, d, n, and p  
32 System.out.println ("\t\t\t\t" + quarters + " quarters");  
33 System.out.println ("\t\t\t\t" + dimes + " dimes");  
34 System.out.println ("\t\t\t\t" + nickels + " nickels");  
35 // print penny if p = 1 else print pennies  
36 if (pennies == 1)   
37 System.out.println ("\t\t\t\t" + pennies + " penny");  
38 else {  
39 System.out.println ("\t\t\t\t" + pennies + " pennies");  
40 }  
41 }  
42 }

Test Runs

1.

89 cents =  
 3 quarters  
 1 dimes  
 0 nickels  
 4 pennies  
  
 ----jGRASP: operation complete.  


2.

16 cents =  
 0 quarters  
 1 dimes  
 1 nickels  
 1 penny  
  
 ----jGRASP: operation complete.  


3.

83 cents =  
 3 quarters  
 0 dimes  
 1 nickels  
 3 pennies  
  
 ----jGRASP: operation complete.  


4.

120 cents =  
 4 quarters  
 2 dimes  
 0 nickels  
 0 pennies  
  
 ----jGRASP: operation complete.  


5.

1 cents =  
 0 quarters  
 0 dimes  
 0 nickels  
 1 penny  
  
 ----jGRASP: operation complete.  


