

# Homework 1

## Computer Science I (20-CS-121) Summer 2007

**Assigned:** Wednesday, 20 June, 2007

**Due:** Wednesday, 27 June, 2007

### 1 Problem Description

Have you ever worked behind a cash register? Once you (or the register) has calculated the total cost for all purchased items, the customer hands you some amount of money. After that, the register tells you how much change the customer is owed, and you take that amount from the register and give it to the customer. If you have worked behind a cash register, then you know what a hassle it can be when you run out of bills/coins. As such, it's best to try and give the customer the fewest amount of bills/coins for their change (i.e. if they are due 99 cents, you wouldn't want to give them 99 pennies).

### 2 Program Description

You are to write a program to help solve this problem. The program should do the following...

1. Prompt the user for the total amount of the charge, in *dollars*. So, if the total amount is \$23.50, the user enters "23.50". If the total is \$0.50, the user enters "0.50".
2. Prompt the user for the amount of cash the customer has given you, again in dollars.
3. If the customer has not given enough cash to cover the total amount, say so and end the program.
4. If the customer has given exact change, say so and end the program.
5. Otherwise, the customer is owed some change...
6. Calculate the amount of change the customer is owed, and report that.
7. Then calculate the fewest possible bills/coins to make the change, and report that. The only bills in the register are \$100, \$50, \$20, \$10, \$5, and \$1. All (common) coins are in the register... quarters, dimes, nickels, and pennies.
8. **NOTE:** If there are fractional cents (e.g. if someone enters a dollar amount with more than two decimal places) you may *drop these values* (ignore them).

### 3 How to submit your program

Once you are finished, you are to email a copy of your source code to me at [ryan.flannery@gmail.com](mailto:ryan.flannery@gmail.com) *before the start of class on Wednesday* and bring a print-out of your source code to class on Wednesday.

Your program *must*...

1. Compile without errors or warnings!
2. Perform the specified task

## 4 Sample runs of such a program

Below are some runs of the program I wrote to accomplish the above task, showing how one interacts with the program. Your program does *not* have to look exactly like the following, but it should accomplish the same thing.

### 4.1 Customer doesn't provide enough cash

In this example, the total cost of the purchase is \$100. The customer only provides 1 cent, and the program alerts the user that not enough money has been provided.

```
Please enter the total cost of the purchase, in dollars: 100
```

```
Please enter the amount of cash received by the customer: 0.01
```

```
The customer has not given you enough money!  
Banish them from your establishment!
```

### 4.2 Customer provides exact change

In this example, the total cost of the purchase is \$5 and the customer provides exactly \$5.

```
Please enter the total cost of the purchase, in dollars: 5
```

```
Please enter the amount of cash received by the customer: 5
```

```
No change due.
```

### 4.3 Customer gives you too much, and is owed change

In this example, the total cost of the purchase is 1 cent...but the customer (flaunting his/her wealth) provides a \$10,000 bill. Sadly, you do not have a "take-a-penny / leave-a-penny" tray where you work.

```
Please enter the total cost of the purchase, in dollars: 0.01
```

```
Please enter the amount of cash received by the customer: 10000
```

```
The customer is owed $9999.99 dollar(s) in change.  
(That's 999999 cents.)
```

```
The fewest number of bills/coins for this amount is as follows...
```

#### BILLS

```
Hundreds: 99
```

```
Fifties: 1
```

```
Twenties: 2
```

```
Tens: 0
```

```
Fives: 1
```

```
Ones: 4
```

#### COINS

```
Quarters: 3
```

```
Dimes: 2
```

```
Nickels: 0
```

```
Pennies: 4
```