

# Assignment 7 – programming

due in class on Wed 6 Nov (40 points)

Before starting this assignment you should review the [Python notes](#), including the [two exercises](#) at the bottom of that page. You may turn in the exercises in addition to this assignment... they may help your score if the assignment doesn't turn out well for you.

This assignment is an **individual activity**. It asks you to write and test a short Python program that computes and displays a simple store receipt. (We can pretend that this input comes via a bar scanner.) The program will then ask for a coupon code, which (if valid) might result in some discount on the price.

Additionally, the receipt should display tax information and subtotals, according to these rules:

1. Any single item that costs over \$10 is subject to a 5% sales tax.
2. Items that are taxed should appear with an asterisk (\*) on the receipt.
3. The receipt must display the taxable and non-taxable subtotals, the tax amount, and the discount amount, if any.
4. Amounts on the receipt do **not** need to be rounded to the nearest cent, padded to two decimal places, or have decimal points aligned.
5. The coupon code `8over20` should give the shopper a 8% discount on orders over \$20 total. (If the total is not over \$20, then the discount is not applied.)
6. The coupon code `15off` will take 15% off the single most expensive item in the order. This one is a little trickier than the previous!

Below are some sample runs of the program. The parts that the user types are indicated by «angle quotes». Your solution should work for these inputs, but also for others according to the rules above. In the first two samples, I just pressed enter to leave the promotional code blank.

## Sample 1

```
Enter price of item 1: «9.99»
Enter price of item 2: «10.99»
Enter price of item 3: «12.50»
Enter promotional code: «»
```

RECEIPT

Item 1: \$ 9.99  
\*Item 2: \$ 10.99  
\*Item 3: \$ 12.5

Non-taxable subtotal: \$ 9.99  
Taxable subtotal: \$ 23.49  
Tax amount: \$ 1.1745  
Total: \$ 34.6545

### Sample 2

Enter price of item 1: «2.50»  
Enter price of item 2: «11.00»  
Enter price of item 3: «1.25»  
Enter promotional code: «»

#### RECEIPT

Item 1: \$ 2.5  
\*Item 2: \$ 11.0  
Item 3: \$ 1.25

Non-taxable subtotal: \$ 3.75  
Taxable subtotal: \$ 11.0  
Tax amount: \$ 0.55  
Total: \$ 15.3

### Sample 3

Enter price of item 1: «8.99»  
Enter price of item 2: «10.99»  
Enter price of item 3: «9.99»  
Enter promotional code: «8over20»

#### RECEIPT

Item 1: \$ 8.99  
\*Item 2: \$ 10.99  
Item 3: \$ 9.99

Non-taxable subtotal: \$ 18.98  
Taxable subtotal: \$ 10.99  
Tax amount: \$ 0.5495  
Total: \$ 30.5195

Discount: \$ 2.44156  
YOU PAY: \$ 28.07794

#### Sample 4

Enter price of item 1: «70.15»  
Enter price of item 2: «64.20»  
Enter price of item 3: «199.99»  
Enter promotional code: «15off»

#### RECEIPT

\*Item 1: \$ 70.15  
\*Item 2: \$ 64.2  
\*Item 3: \$ 199.99  
  
Non-taxable subtotal: \$ 0.0  
Taxable subtotal: \$ 334.34  
Tax amount: \$ 16.717  
Total: \$ 351.057  
Discount: \$ 29.9985  
YOU PAY: \$ 321.0585

#### Notes and hints

7. Start with the simplest pieces of the functionality you can, and get them to work before adding more. For example, just ask for the price of *one* item, and determine whether or not it is taxable. Run and test that program before moving on.
8. We need the user to input numbers with decimals, which are called ‘floats’ in most programming languages. Recall the syntax of the input statement. The *prompt* is inside the quotes. The variable `price1` will then contain whatever number the user typed.

```
price1 = float(raw_input("Enter price of item 1: "))
```

9. The problem description refers to percentages, such as a 5% tax rate, and a 10% discount. You cannot use the percent sign directly in Python, instead we have to divide by 100 and use that decimal number. So 5% is written as `0.05`, as in:

```
tax = taxable_sum * 0.05
```

10. Try to use sensible variable names, such as `tax`, `total`, `discount`, etc. Names cannot have spaces in them – instead we use the ‘underscore’ character, as in `nontax_sum`.

Email your `.py` file to me as an attachment by class time on the due date. Use a comment at the top to write your name and assignment info:

```
# Python receipt printer, by YOUR NAME  
# November 2013
```