

---

## CMSC 201 Fall 2016

### Homework 2 – Pseudocode to Code

**Assignment:** Homework 2 – Pseudocode to Code

**Due Date:** Wednesday, September 21st, 2016 by 8:59:59 PM

**Value:** 40 points

**Collaboration:** For Homework 2, collaboration is not allowed – you must work individually. You may still come to office hours for help, but you may not work with any other CMSC 201 students.

Your collaboration statement should state that

**Collaboration was not allowed on this assignment.**

**UPDATE:** Questions 2 and 3 have been updated. Please see the individual questions for details.

Make sure that you have a complete file header comment at the top of each file, and that all of the information is correctly filled out.

```
# File:      FILENAME.py
# Author:    YOUR NAME
# Date:      THE DATE
# Section:   YOUR DISCUSSION SECTION NUMBER
# E-mail:    YOUR_EMAIL@umbc.edu
# Description:
#   DESCRIPTION OF WHAT THE PROGRAM DOES
# Collaboration:
#   COLLABORATION STATEMENT GOES HERE
```

Homework 2 is designed to help you practice translating pseudocode and flowcharts into Python code. Remember to enable Python 3 before running and testing your code:

```
scl enable python3 bash
```

## **Instructions**

For each of the questions below, you are given an algorithm, expressed in either pseudocode or a flowchart. From this information, you must implement a working solution in Python. For this exercise, you will only need to use concepts we have discussed in class such as variables, expressions, casting to an integer, and `print()`.

The pseudocode and flowcharts may combine multiple lines of code into one step, or they may split something that would take a single line of code into multiple pieces. Think carefully about what the overall goal of the algorithm is before you begin coding.

**At the end, your Homework 2 files must run without any errors.**

## Questions

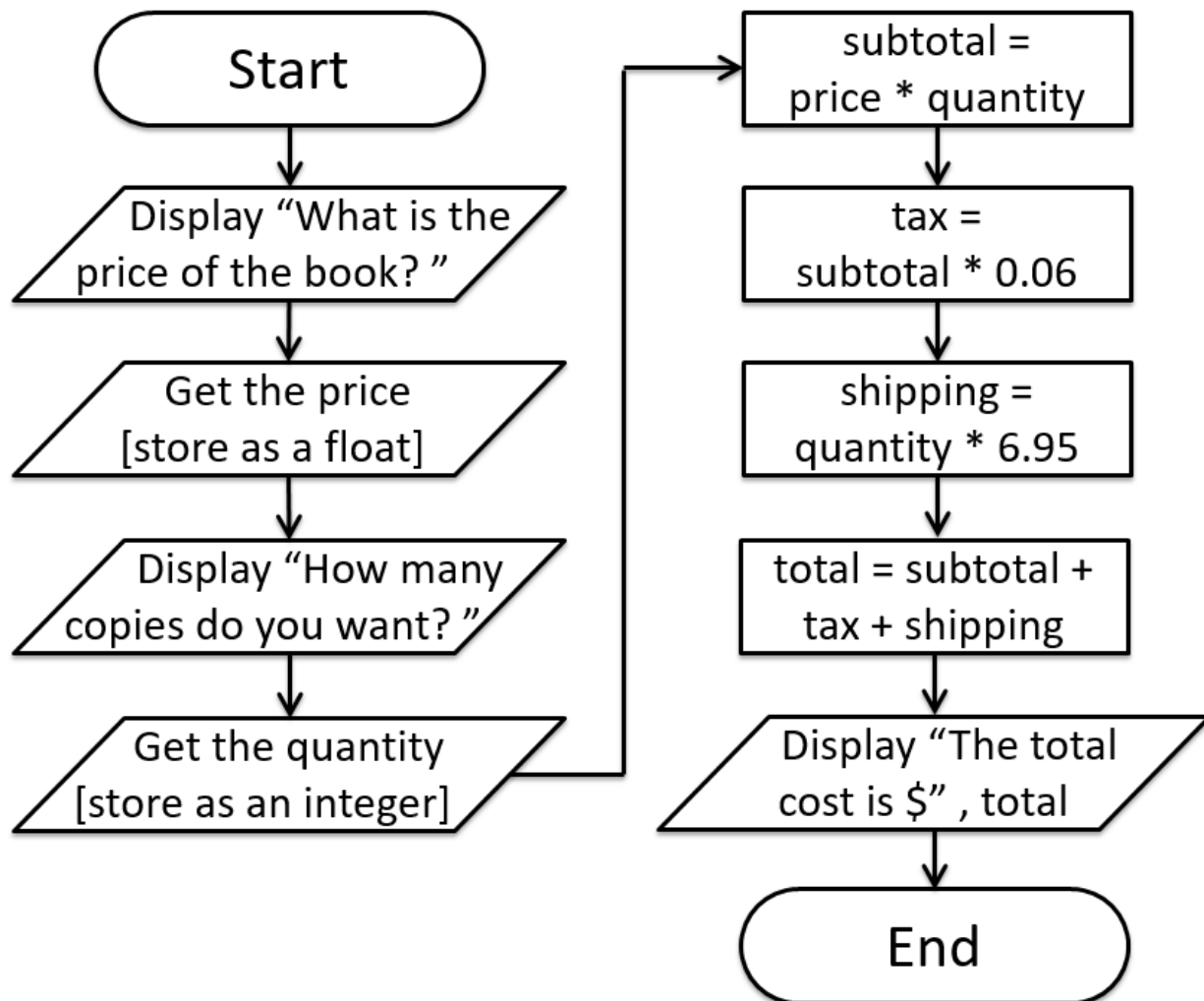
Each question is worth 9 points. Following the directions is worth 4 points.

### Question 1

Write your program for Question 1 in a file called `hw2_part1.py`.

This algorithm, shown in a flowchart, calculates the total cost to buy and ship a large order of books from an online bookstore.

Translate this flowchart into a Python program.



## Question 2

Write your program for Question 2 in a file called `hw2_part2.py`.

This algorithm, shown in pseudocode, describes how to extract the number of dollars and number of cents from a price when the price is given as a float. For example, a price of 5.99 yields a value of 5 for the dollars and 99 for the cents.

Translate this pseudocode into a Python program.

```
Display "What is the price?"
Store the value as a float
```

```
Cast the price to an integer, and store the result in variable
(HINT: Think about what this number represents)
```

*(UPDATED INSTRUCTIONS BEGIN HERE)*

```
Subtract the number of dollars from the price, and multiply the
result by 100 to get the number of cents
```

```
Cast the numbers of cents to an integer, and save the result
```

*(UPDATED INSTRUCTIONS END HERE)*

```
Display "The number of dollars is: " and the number of dollars
```

```
Display "The number of cents is: " and the number of cents
```

For this algorithm, the names of the variables are not given to you. You should choose meaningful variable names.

Casting a variable to an integer is covered in detail in Lecture 4. A short example is shown below:

```
myFloat = 7.76
myInt   = int(myFloat)
# The value of myInt is now 7 (an integer)
```

### Question 3

Write your program for Question 3 in a file called `hw2_part3.py`.

This algorithm, shown in pseudocode, is a very simple implementation of a basic shopping list application.

Translate this pseudocode into a Python program.

```

Display "What would you like to buy first?"
Store the value as a string called item1

Display "You are buying [item1]" (UPDATED INSTRUCTION)
Display "How many would you like to buy?" (UPDATED INSTRUCTION)
Store the value as an integer called amount1

Display "What would you like to buy second?"
Store the value as a string called item2

Display "You are buying [item2]" (UPDATED INSTRUCTION)
Display "How many would you like to buy?" (UPDATED INSTRUCTION)
Store the value as an integer called amount2

Display "What would you like to buy third?"
Store the value as a string called item3

Display "You are buying [item3]" (UPDATED INSTRUCTION)
Display "How many would you like to buy?" (UPDATED INSTRUCTION)
Store the value as an integer called amount3

Calculate how many total items the user is buying, and store in a
variable called totalNumber

Display "You are buying [totalNumber] items:"
Display "[amount1] [item1]"
Display "[amount2] [item2]"
Display "[amount3] [item3]"

```

When the algorithm shows `[item1]` or something similar in the text to display, you should write your code so that it asks about the specific item. You can see an example of what the program might look like when run on the following page.

Here is some sample output, with the user input in blue.

```
What would you like to buy first? bananas
You are buying bananas
How many would you like to buy? 5

What would you like to buy second? ramen
You are buying ramen
How many would you like to buy? 12

What would you like to buy third? milk
You are buying milk
How many would you like to buy? 2

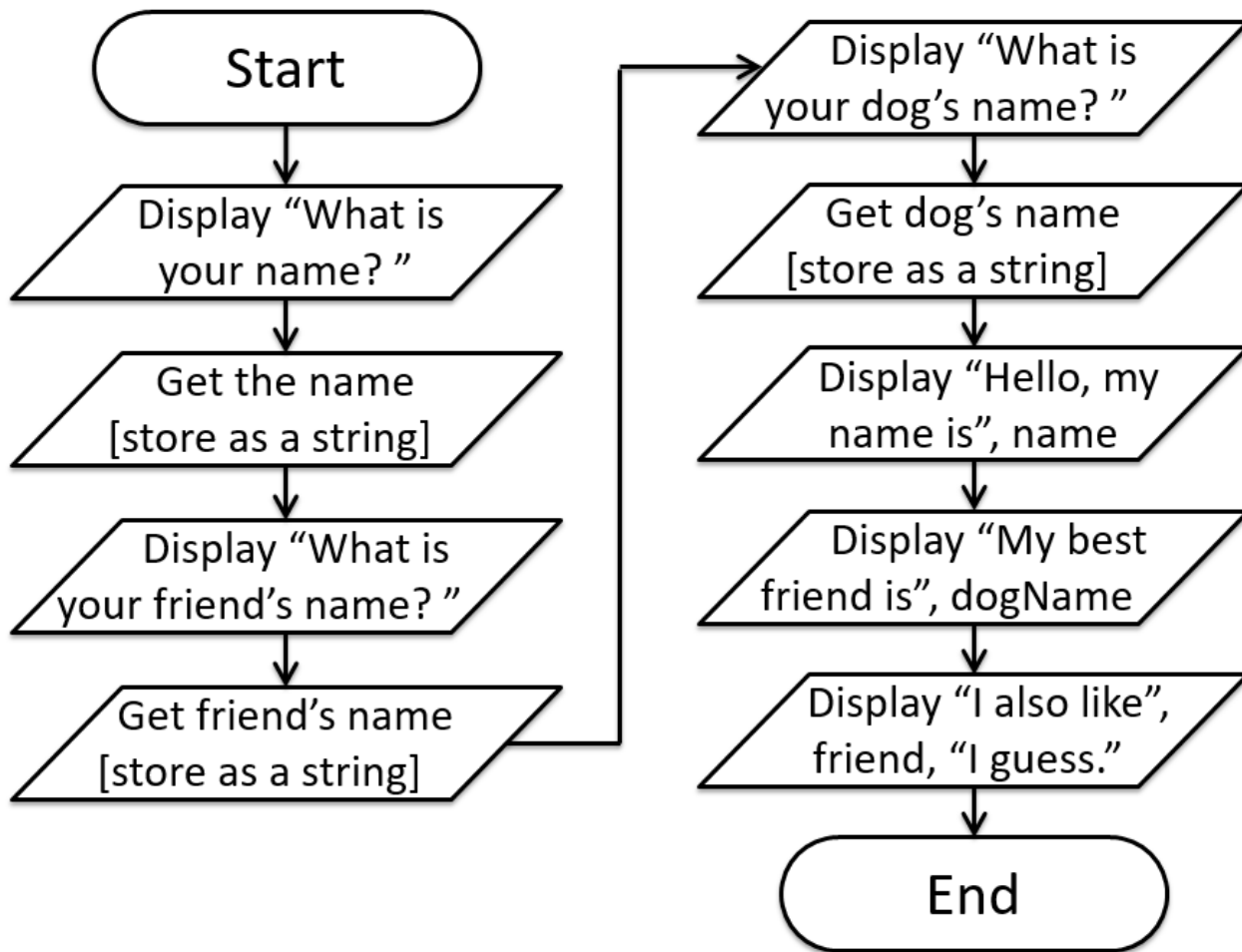
You are buying 19 items:
5 bananas
12 ramen
2 milk
```

Question 4

Write your program for Question 4 in a file called `hw2_part4.py`.

This algorithm, shown in a flowchart, interacts with the user in a very simple way, asking them some information about themselves, and then repeating it back in a slightly different way.

Translate this flowchart into a Python program.



## Submitting

NOTE: How to submit is covered in Lab 1. If you have not completed Lab 1, you should do so before completing this part of the homework.

Once your `hw2_part1.py`, `hw2_part2.py`, `hw2_part3.py`, and `hw2_part4.py` files are complete, it is time to turn it in with the `submit` command. (You may turn in individual files as you complete them. To do so, only `submit` those files that are complete.)

You must be logged into your GL account, and you must be in the same directory as your Homework 2 python files. To double-check this, you can type `ls`.

```
linux1[3]% ls
hw2_part1.py hw2_part2.py hw2_part3.py hw2_part4.py
linux1[4]% █
```

To submit your Homework 2 python files, we use the `submit` command, where the class is `cs201`, and the assignment is `HW2`. Type in (all on one line) `submit cs201 HW2 hw2_part1.py hw2_part2.py hw2_part3.py hw2_part4.py` and press enter.

```
linux1[4]% submit cs201 HW2 hw2_part1.py hw2_part2.py
hw2_part3.py hw2_part4.py
Submitting hw2_part1.py...OK
Submitting hw2_part2.py...OK
Submitting hw2_part3.py...OK
Submitting hw2_part4.py...OK
linux1[5]% █
```

If you don't get a confirmation like the one above, check that you have not made any typos or errors in the command.

You can check that your homework was submitted by following the directions in Homework 0. Double-check that you submitted your homework correctly, since **an empty file will result in a grade of zero for this assignment.**