*#1*

CPT 167—Intro to Programming Logic

Program #1

**PROGRAM DESCRIPTION**

This is going to be a rather simple program—I know, easy for me to say—that is not intended to push the envelope of digital computer technology, but rather to give you some practice using a variety of inputs and outputs and a rather straightforward calculation or two. You will find much planning for this program in an addendum, so don’t forget to study that also!

What we’ll do with this program is ask the user to enter some simple data, do a few calculations, and provide answers. We’ll pretend that the user is a customer who has just walked into our school’s bookstore, and we want to be very user-friendly. So, the first question we’ll ask is the customer’s first name. Then, for the rest of the dialog, we’ll use their name as we get more information and display our final answers. For example, if I were the customer and inputted my first name as Carol, then I would expect the next question to be phrased something like this:

*OK, Carol, thanks for stopping here for your supplies. Now, I’m just wondering, how long have you been attending the school? Just enter the number of years, please:*

In other words your program will be acting as the sales person. We’ll get as much information as we need from the customer, do some simple math, and then answer all their questions. One of the reasons we need a program is that everything in the store has been discounted by 37% today, so business is going to be brisk! Lots of great stuff that every student needs is going out the door today, and this program will determine the final cost of anything we will sell. Also, one factor will slightly complicate the plan. Your user is rather short on cash, and therefore wants to spend any money wisely. As a result, you will need to compute and display the sale price on the item they want, and then ask for how many they want to purchase. Make sure you don’t ask for the count of how many they want until after you show them the new, lower price.

You will need to create a data dictionary, a pseudocode plan, and a Java project to solve this problem. Your basic plan should be this: ask for as much of the input data as possible; compute the new, lower price for one item; display that price; ask for how many the want of that item; compute the total cost for as many of these items as the customer wants to buy; and finally display all the information on screen for the customer. Try to make your program as user-friendly as possible.

After you’ve completely planned out the program and are ready to code, start up Eclipse and create a new project and HTML file. As with Program #1 be very careful to follow the instructions in the “Intro to Java…” addendum when getting the actual program started. The name of the project you create should be this: ***name* Program 1 Project**, where *name* is your last name. When you add the HTML file, give it some sort of meaningful name for this program. As explained in the Coding Standards document, I will be grading these names for correctness: source code filenames should not be haphazard, they should tell something about the program itself, what’s its purpose is.

**SPECIFIC DIRECTIONS**

Ask the user of your program (be respectful!) to enter the following information. When prompting for each value, try to think how a nonprogrammer would need to be asked for the information: don’t assume your user is a computer genius! Input the following, in this order:

• customer’s first name (Then, use their name for the rest of the input questions.)

• how long they have been attending this school (in years only)

• name of the item they want to purchase

• original price of one item (Yes, we’ll just trust the customer for this price!) As

described above, then compute and display the discounted

price for that one item.

• how many of these they want to purchase (You need more than one, don’t you? )

Say, for example, they are interested in buying several blue pens to take notes for the entire semester. We’ll ask for the price for one blue pen, then determine the discounted cost for one of them, display that, get how many they want, and then figure out the total cost of all of the pens they want to buy.

This shopping season, everyone is getting a 37% discount. (Don’t ask why…some genius came up with this figure!) So, make sure you use that to come up with the discounted price for any item they purchase. Once all computations are done, display all the following: first, all data that was inputted at the start; then, the discount percentage given for the sale; next, display the new price for one of the items (again); and last, display the total cost of purchasing however many of the items the customer wants. Your output should be just as user-friendly as your input prompts.

Suppose again that the customer’s name is Carol, and Carol wants to purchase 6 blue pens, which ordinarily sell for $2.70 apiece. Your final output might look something like this:

OK, Carol, you want to buy 6 blue pens today. Previously, these sold for $2.70. But they are on sale today for a 37% discount, so their price today is only $1.70. The total will come to $10.21. Since you have been attending the school for 2 years, you have obviously found it necessary to take a lot of notes. Good luck with your class work this semester!

Here, I have underlined the values that you would have stored in variables just to set them apart. I would not expect you to underline anything in your output. These are values that you would either have inputted or calculated. But also note that every single thing that we inputted is also a part of the output, as are the values we calculated, and we even quoted the discount percentage, which was a constant. Everything is on display!

(Note: the values may print out a bit “funny”, with decimal values you don’t want. Don’t worry about that just now: we will fix that in the next program. You will learn how to control how many decimal places appear in your output.)

For this program I do not want you to display any of the final answers until you have completed the calculations. For any sale we always like to sum up the entire transaction. That’s called being “user-friendly” and that’s what our bookstore is noted for!

Zip your entire project directory together for turn-in, as explained in the Coding Standards handout.

**TURN IN:** a data dictionary for this program, a pseudocode plan, and the zipped project directory. Submit everything using the Dropbox submission feature in D2L.