Assignment 03: Getting Iffy With It

Due: Friday, October 6, 2017 @ 11:59pm

We're going to get you branching out now - creating forks in the road. We'll be starting things easy-ish this week and then ramping up the difficulty - and usefulness - in our next run at conditionals.

You must use simple if/else statements for *every* challenge in this problem set. This means **no**:

- nested ifs
- ifs with compound conditionals (like && or ||)
- no else ifs

Oh, one more thing. If you had trouble getting your laptop (Mac or PC) working with the automated testing on the first assignment, please go talk to Jordan Pratt (jpratt@mtroyal.ca), whose office is in B103 – he can make sure you're good to go for the testing in this go-round.

Canada Post Challenge

Mail Time!

I had to send a package recently to a friend in Grande Prairie. Brought the thing up to the counter and it was weighed, measured, stickered and placed with care into the loving arms of our national postal service. I said to myself, "Self - this would make a helluva good coding challenge!" Lucky you.

After digging around a bit on the Canada Post site, I found this document that has the Canada Post prices for 2017. On page 5, it basically says to calculate a shipping price, you need to:

1. "Find the applicable rate code for your packages origin and destination."
2. "Refer to your price sheets to determine the shipping price according to the apppropriate [sic] rate code, weight and service."

Since making a full-blown price calculator is **way** beyond cruel, we're going to something more stripped down: a little app that, when given the dimensions and weight of a package, tells you what row to look up in the Regular Parcel Prices table (starting on page 10 in the price doc mentioned earlier). And it will give you something called the Forward Sortation Area as well.

MAKE THIS HAPPEN:

Write code that:

1. Prompts the user for and obtains the width of the package in cm.
   - You may assume the user enters a valid positive number. This number may have a decimal.
2. Prompts the user for and obtains the height of the package in cm.
   - You may assume the user enters a valid positive number. This number may have a decimal.
3. Prompts the user for and obtains the length of the package in cm.
   - You may assume the user enters a valid positive number. This number may have a decimal.
4. Prompts the user for and obtains the weight of the package in kg.
   - You may assume the user enters a valid positive number. This number may have a decimal.
5. Prompts the user for and obtains the destination postal code.
   - You may assume that the user enters a postal code...but they may forget the space and/or forget to capitalize everything.
6. Outputs one of the two following sentences:
   - "`The shipping charge will be based on package weight.`"
   - "`The shipping charge will be based on the volumetric equivalent of weight.`"
   - You **must** include these sentences **exactly** as written, or the automated tests will fail!
6. Outputs the weight row to use in the price sheet for regular parcels.
   - For example, "`Look up weight in row 10.5`"
   - The automated tests only care about the number for this output. The format of these numbers must match what you see in the first column of the Regular Parcel Prices table.
7. Outputs the first 3 characters of the destination's postal code - this is called the *Forward Sortation Area (FSA)*. (This would be used by the user of the app to figure out what column in the table to use.)
   - For example, "`Forward Sortation Area: T3E`"
   - The automated tests need the FSA to be in all caps.

NOTES:

- You'll have to read section 3.4 (*Shipping charges and weight*) of Canada Post's ABCs of Mailing on how to calculate the volumetric equivalent of actual weight.
    - From the same section, notice that when calculating your weight row, you need to round "to the nearest 0.500". But in the price tables given, prices are rounded to the nearest 0.5, so we'll use that instead.
- Also from the same section, notice that the minimum weight/VE row possible is 0.75...this is different from the other rows, where you're rounding to the nearest 0.5. You have to deal with this.
- If the package weight and VE equivalent of weight are the same, then the shipping charge should be based on package weight.

Superfun Quiz Challenge

Pop quiz, hotshot!

I think that random trivia is fun. If you combine that with the utter joy of multiple-choice questions, you get the Superfun Quiz™:

Entomology is the science that studies
A. Behavior of human beings
B. Insects
C. The origin and history of technical and scientific terms
D. The formation of rocks

Eritrea, which became the 182nd member of the UN in 1993, is in the continent of
A. Asia
B. Africa
C. Europe
D. Australia

Which of the following is not one of the bots on Mystery Science Theater 3000?
A. Jambot
B. Gypsy
C. Tom Servo
D. Crow T. Robot

Germany signed the Armistice Treaty on ____ and World War I ended
A. January 19, 1918
B. May 30, 1918
C. November 11, 1918
D. February 15, 1918

The headquarters of the United Nations are situated at
A. New York, USA
B. Hague (Netherlands)
C. Geneva
D. Paris

What utensil serves as the rallying cry of The Tick?
A. chopstick
B. spoon
C. fork
D. butter knife

What is the theme of the board game Die Macher?
A. sculpting
B. car racing
C. German politics
D. castle building

Whew, I hope you can handle the fun!

MAKE THIS HAPPEN:

Write code that:

1. For each of the questions above:
   - Display the question.
   - Display the answers.
   - Ask the user for an answer.
2. As the questions are answered, the program must keep track of the number of correct answers.
3. Also keep track of the number of C answers that the user entered.
4. After all of the questions have been answered, output something like one of the following sentences:
   - `"You have passed the quiz!"`
   - `"You have failed the quiz."`
   - The automated tests will look for “pass” or “fail” in your output.
5. If the user entered all C answers, the program should also print:
   - `"All Cs, eh? That’s how I passed music theory last year."`
   - The automated tests will look for “Cs” in your output.

NOTES:

- You'll have to determine the correct answers to the quiz.

- The user passes the quiz if they have at least 5 correct answers.

- The user must be able to enter their answers as an uppercase or lowercase letter.

- The answers must be stored in char variables and compared to another char in the `if` statements.

Marking

This assignment will be marked using two automated tools as well as manually by your instructor.

Here is the marking rubric:

Score = (8 x Automated Correctness Checks) +
(4 x Automated Style & Best Practice Analysis) +
(8 x Design)

Automated Correctness Checks (*Does your code do what it was asked to do?)*

- 5: Both challenges pass.
- 4: One challenge passes and another passes half the tests.
- 3: One challenge passes and the other passes less than half the tests.
- 2: One challenge passes at least half the tests and the other passes less than half the tests.
- 1: At least one test passes.
- 0: One or more compilation errors.

Automated Style & Best Practice Analysis (*How closely does your code follow the automated rules you are given?*)

- 5: No errors detected.
- 2: One or two **types** of errors detected.
- 0: Three or more **types** of errors detected.

Design (*How closely does your code approach the design ideals of clarity, efficiency, and maintainability? How much care have you put into your work?*)

- 5: Excellent. Essentially no room for improvement.
- 4: Very good. Minor room for improvement.
- 3: Good. Some room for improvement.
- 2: Fair. Ample room for improvement.
- 1: Poor.
- 0: No attempt.

Cucumber

Cucumber is a testing tool. It uses a list of requirements to describe what some code should do. These tests have been predefined and are included with the starting code for the assignment. These tests will be used to determine the “Automated Correctness Checks” portion of your mark.

SonalLint

SonarLint is a code quality analysis tool. It tests that code meets some minimum set of standards. The output of SonarLint will be used to determine the “Automated Style & Best Practice Analysis” portion of your mark.

Submission Instructions

Create a folder named Asg03_*`Lastname_Firstname`* and place your BlueJ project (the contents of the `solution` folder) in the folder. Submit your source code to the submit folder (`I:\Labs\CompSci\Submit\1501\00x`). If you are submitting your files via the web interface from off campus, you will need to compress your files into one `.zip` file.