CSCA20 - Lab 4

Selection

Learning Objectives

This lab focuses on selection, building up your skills with using if statements, and also combining them with loops to build more interesting/powerful tools.

Prelab

We've provided you with some starter code that prints a menu and asks the user to choose an option. Before your tutorial, you should get this code running, and write code that will print the action needed based on that input (e.g., if the user chose `T`, it should print `Calculating total`)

Demonstration & Evaluation

Successfully completing this lab will demonstrate competency in Variables, Loops and Selection. There will also be an opportunity to demonstrate mastery in user Input/Output. You do not need to complete the mastery portion to demonstrate competency in the other areas.

The Scenario

The Union for Tracking Seaborn Contaminants (UTSC) has asked you to build a tool to track the number of kilograms of plastic their ocean cleanup rigs clean on a daily basis [1]. They want to be able to add the kilograms cleaned each day, calculate the total amount removed over certain time periods, and the number of days that their rigs were pushed beyond their approved limits by collecting too much plastic.

Menu

The starter code presents a simple menu that gives the user options for entering data, retrieving information, or quitting.

If the user enters any other letter, the code should say `invalid input:  please try again` and present the menu again.

Adding Data

Your code should prompt the user with the date and ask for input. e.g., `Enter kg of plastic removed on day 17:`

Calculating Totals

Your code should prompt the user to select `A` for the total removed in all days entered, or `R` for the total removed in a range of days. If the user chooses `R`, they should then be prompted for the start and end dates of that range. The user should then be presented with the total kg removed during the specified time.

Calculating Overload Days

Your code should prompt the user to select `A` for all overload days, or `R` for overload days within a range of days. Then the user should be prompted for the threshold (max kg rigs are meant to collect) and then be presented with the total number of days where the collected amount exceeded the threshold during the specified time.

Mastery

You can demonstrate mastery of user input/output in this lab by setting up your menus and responses such that the user gets sensible feedback for non-sensible input. Your code should not crash, and should prompt the user for better input if they give bad input. For example, if they choose menu options that don't exist, or enter invalid dates or words instead of numbers for amount of plastic on a given day.

[1] This is a real project... just not the real name for it, check out `https://theoceancleanup.com/oceans/` for more info

Hints

Here are a few hints that might help you with this assignment:

- Break the problem down: don't worry about calculating totals while you're worried about getting user input. Do all the user input parts first and just leave a comment saying `#calculate total here` or `#calculate overload days here`, and come back to those separately
- Use comments: this is a big lab, it can be easy to get lost. Plan out your code with comments first, and make sure you use your comments to remind yourself what you're doing where
- Think about your if statements: you can do this just using lots of ifs, but your life will be easier if you set up appropriate if/elif/else structures

Extra Practice

If you finish early and want to practice more, here are a few things you can try:

- Completing the mastery portion isn't a lot of work, but it is time consuming. Making idiot proof code is difficult, because there are a lot of very resourceful idiots out there.
- Try to add additional options such as retrieving the maximum or minimum amount of plastic removed in a given date range
- Try to improve the user input experience, so that the user can enter several days at once (e.g., "enter plastic for next day or 'E' to exit")