

Python Bootcamp

Week 2 - Challenges

Personal Unit Converter Challenge

You will create an interactive program which will act as your own personal unit converter. Your program will do the following things:

1. When the program starts it should ask for the user's name and then present a welcome message (using the user's name!)
2. It will then present the user with a list of possible conversions it can perform and ask for the user to choose an option.
3. The user then makes a selection and the program will ask for a number to convert.
4. The program will then make the conversion and print it out.
5. The program will then present a closing message (again using the user's name)

Your program will need to be able to perform at **least 5 different** conversions. In your code, you will write a function for each of your converters. Based on the user's input you will call the appropriate function. You are free to choose whatever conversions you would like but you must include a "percent to letter grade" converter in your program. Use the following (simplified) ranges for this conversion:

Range	Grade
[80, 100]	A
[70, 80)	B
[60, 70)	C
[50, 60)	D
[20, 40)	E
[0, 20)	F

A note on this range notation: Depending on the type of bracket used when describing a range, the range will either be **inclusive** or **exclusive** at that end. An inclusive end is represented using a square bracket and means that this number **is included** in the range. An **exclusive** end is represented using a round bracket and means that everything up to **but not including** this number is included in the range.

For example, the A range is from 80% - 100%, including both 80% and 100%. On the other hand, the C range is from 60% up to but not including 70% (ie. any number in the sixties).

Here is a few examples of what running your program could look like (user input is highlighted to help illustrate the example):

```
Please enter your name: cat
Hello Cat, welcome to your personal unit converter.
Please choose which conversion you would like to perform:
1 - convert cm to inches
2 - convert percent to letter grade
3 - convert cups to ml
4 - convert grams to ounces
5 - convert fahrenheit to celsius
Choice: 1
Value in cm to convert: 10
10 cm = 3.937007874015748 inches
Goodbye Cat.
```

```
-----
Please enter your name: dog
Hello Dog, welcome to your personal unit converter.
Please choose which conversion you would like to perform:
1 - convert cm to inches
2 - convert percent to letter grade
3 - convert cups to ml
4 - convert grams to ounces
5 - convert fahrenheit to celsius
Choice: 2
Percent to convert to letter grade: 67.1
67.1% = C
Goodbye Dog.
```

Ship Boxes of Books Project.

Overview

In this project, you will demonstrate your ability to take an English problem statement and convert it into a Python solution.

Objectives

Building on everything you have learned so far in this boot camp, the objectives of this project are to demonstrate an understanding of the following:

- Input/output
- Variables
- Arithmetic expressions
- Division and modulo
- if-else statements
- Equality and relational operators
- Nested if-else statements
- Code blocks and indentation

Description

There are two parts to this project:

- **Part 1** is your pseudo code/algorithm and will be uploaded to GitHub.
- **Part 2** is to complete the program in Python.

Packing boxes with books that a customer has ordered. Remember that there is only one book per layer, that is, you do not have to worry about the length and width of the box or the book, only the height. Each box is packed with a stack of books that are all the same size. You have boxes that are all the same width and length that fits the book exactly, but they are 2 different heights. It should be well documented.

You will:

- Ask the user for the height of the big box, and the height of the small box. The height should be entered as a single number in inches, for example, 8.5.
- Ask the user for the height (thickness) of the books, also in inches.
- Ask the user for the number of books ordered.

You may use whatever prompts you to want, but the input must be in the order listed above.

You will then use this information to determine how to ship the order, in how many boxes of each kind.

Your goal is to ship the fewest number of boxes. Your algorithm should try to fill the larger box if it can be filled, and should not partially fill a large box when the smaller one will do.

You will need to print out the number of each kind of box that will be needed. In the special case of a single box, use exactly the following phrases, as appropriate:

```
Ship 1 small box or Ship 1 small box
```

In the case of more than one box, print the total number of boxes shipped, and the number of each kind, starting with large (if any), then small (if any). Do not indicate any instance of 0 boxes of a particular type. For example, if you will ship 2 large, 0 small, you should print out something like:

```
Shipping 2 boxes  
2 large
```

IMPORTANT! There are two components to this project: pseudo code of your algorithm, and the final program.

Before you start to code, and even before you start to develop pseudo-code, make sure you thoroughly understand the problem. Think about how you as a human would solve this problem. Use concrete numbers as examples. Go through several scenarios of book orders to see what the correct number of boxes would be. Make sure you have done this thought exercise and have it written down before you start to code. If you ask for help in developing the pseudo-code, your Team Leader may ask to see your thought processes and ask you for some example solutions. If you can't solve the problem with your own brain, you will never be able to come up with a Python program to solve it. If you ask for help with the program, your TL may ask to see your pseudo-code first to make sure you have already put significant thought into solving it on your own.



TILL YOU DROP!