

## Computer Project #2

### Assignment Overview

This assignment focuses on the design, implementation and testing of a Python program to solve an arithmetic puzzle using repetition and selection.

It is worth 20 points (2% of course grade) and must be completed no later than 11:59 PM on Monday, September 19.

### Assignment Deliverable

The deliverable for this assignment is the following file:

`proj02.py` – your source code program

Be sure to use the specified file name and to submit it for grading via the **handin system** before the project deadline.

### Assignment Background

**Puzzle:** Write a program that adds all the digits in an integer. If the resulting sum is more than one digit, keep repeating until the sum is one digit. For example, the number 2345 has the sum  $2+3+4+5 = 14$  which is not a single digit so repeat with  $1+4 = 5$  which is a single digit.

### Assignment Specifications

Your program will prompt for one natural number, i.e. a positive integer. If a zero or negative integer is input, your program will re-prompt until a positive integer is input. Your program will calculate the sum of the digits of the number and print the arithmetic expression (see sample below). If the sum is not a single digit, repeat.

### Assignment Notes

Divide-and-conquer is an important problem solving technique. In fact, inside of every challenging problem is a simpler problem trying to get out. Your challenge is to find those simpler problems and *divide* the problem into smaller pieces that you can *conquer*. For this project we will lead you through those smaller problems.

1. Leave the error checking to the end by assuming for now that a correct value will be input.
2. Since we need a program that finds the sum of the digits of a number, let's begin with a simpler program that takes an integer  $N$  as input and prints the digits. You can use the mod (remainder) operator `%` to extract the least significant (rightmost) digit of a number. Since the numbers are base 10 you can use  $N \% 10$ . (Try that in the Python shell.) Print

that digit. To get the next digit you next must remove the first digit from the number. How? Remainder produced the digit so the quotient operator “//” will remove it, i.e.  $N=N//10$ . (Try that in the Python shell.) Now put that in a loop. Since you do not know how many digits are in the number, you need to use a *while* loop. What will your Boolean expression be for the while loop? How will you know that you are done? What value will N be that indicates that you are done? Test your program to ensure that it can print the digits of any number of any length.

3. Now that you can print the digits, print them in one line with a space in between using the *end* argument in the print function: `print(x, end = " ")`. Similarly, you can put a “+” sign after each one, but you need an *if* statement to print something different after the last one—since you have figured out how to stop your *while* loop use that same Boolean to decide which is the last number so you don’t print a “+” sign after it.
4. Now you need to sum the digits. Initialize a variable to zero and each time you extract a digit add it to your sum: `digit_sum = digit_sum + 1` (or you can use `digit_sum += 1`). Print that sum (you know when you are at the end because you just figured that Boolean out in the previous step). Test your program on a variety of input.
5. Now that you have a program that can find and print the sum of any natural number you can consider the next part of the original problem: continue the process until the sum is a single digit. Since you don’t know how many times you will have to do this, you need to wrap another *while* loop around your current loop. (Within the Spyder editor you can highlight your current loop and under the “Edit” drop down menu you can select “indent” to indent the highlighted code.) Next you need to figure out a Boolean to control this outer loop: how do you know that your sum is a single digit? Test your program.
6. With a working program you can now add the error correction needed for the input. A *while* loop works nicely: keep looping until a correct number is input. What Boolean is appropriate to figure out if an integer is a natural number? (Note that at this point in the course we don’t know how to handle input that isn’t an integer. On such input your program will crash and that is fine for now. If you want to look ahead to figure out the proper way, you need to use an *exception*.)

## Suggested Procedure

- *Solve the problem using pencil and paper first.* You cannot write a program until you have figured out how to solve the problem. This first step can be done collaboratively with another student. However, once the discussion turns to Python specifics and the subsequent writing of Python, you must work on your own.
- Use Anaconda to create a new program. Use the required file name (`proj02.py`).
- Write a simple version of the program, e.g. one wind chill calculation without input. Run the program and track down any errors.
- Use the **handin** system to turn in the first version of your program.

- Cycle through the steps to incrementally develop your program:
  - Edit your program to add new capabilities.
  - Run the program and fix any errors.
- Use the **handin** system to submit your final version.

## Sample Output

```
In [38]: runfile('/Users/enbody/Documents/cse231/FS16/Projects/Project02/Admin/summ.py',
wdir='/Users/enbody/Documents/cse231/FS16/Projects/Project02/Admin')

Input an integer: 35
5 + 3 = 8

In [39]: runfile('/Users/enbody/Documents/cse231/FS16/Projects/Project02/Admin/summ.py',
wdir='/Users/enbody/Documents/cse231/FS16/Projects/Project02/Admin')

Input an integer: 2345
5 + 4 + 3 + 2 = 14
4 + 1 = 5

In [40]: runfile('/Users/enbody/Documents/cse231/FS16/Projects/Project02/Admin/summ.py',
wdir='/Users/enbody/Documents/cse231/FS16/Projects/Project02/Admin')

Input an integer: 98765678912398
8 + 9 + 3 + 2 + 1 + 9 + 8 + 7 + 6 + 5 + 6 + 7 + 8 + 9 = 88
8 + 8 = 16
6 + 1 = 7

In [41]:
```