Intro to Python – Lesson 15 & 16

Today we want to look at input validation. This means we check each input to make sure it is valid before we continue and make the user re-enter the value if it is not valid. Check out the following videos before the start of class and we will discuss and look at examples in our class discussion.

https://www.youtube.com/watch?v=-8bDdrZhj_k (What is input validation?)
https://www.youtube.com/watch?v=jlOyXjfxEOs (Use try / except / else to validate)

For string values, the validation is a bit simpler since there is no conversion to a number that needs to take place. I will still use a loop, and just an if statement to check for valid input. For example, if you are entering a value to see if a job is after hours (Y / N) – the user must enter a value – cannot be blank – and the value must be a Y or N.

```
while True:
    AfterHours = input("Was this job completed after hours (Y / N): "))

    if AfterHours == “”:
        print("After hours cannot be blank - Please enter a Y or an N.")
    elif AfterHours,upper() != “Y” and AfterHours.upper() != “N”:
        print(“After hours must be a Y or an N – Please re-enter.”)
    else:
        break
```

For each numeric input that needs to be validated I will add a loop and the try / except / else for each input – this basically check and makes sure the value is a number. A full example of a validation is included below – in this case I input the number of hours – make sure it is a valid integer in the except – and check the range in the else. Note if there is no range check the else: will contain only the break statement.

```
while True:
    try:
        Hours = input("Enter the number of hours to complete the job: ")
        Hours = int(Hours)  # This is where the try / except kicks in.
    except:
        print("Number of hours is not valid – do decimal allowed - please re-enter.")
    else:
        if Hours <=1 or Hours >=20:
            print("Number of hours must be between 1 and 20 - Please re-enter.")
        else:
            break
```

Try a few of these examples and we will discuss in class.

- Billy Bob Bike Rentals requires a program to process the bike rentals he has made for the day. Required are the customer's name (must be entered), a phone number (must be entered), a code for the type of bike rented (T for 12-Speed, M for Mountain, B for a bicycle built for 2), the number of bicycles rented (must be between 1 and 3), a credit card number, and the expiry date. Add a calculation or two and some output. Prompt the user if they want to continue with a Y/N response.

- A local Used Car Company requires a program to process used car sales. Input will include the customer's name (must be entered), the phone number (must be entered), the car year, make and model (one variable – must be entered), the car price (between $1,000.00 and $10,000.00), and a trade in allowance (cannot exceed $10,000.00). Add a calculation or two and some output. Allow the program to repeat until the user enters "END" for the Customer name.

- A local company called The Snuggly Company sells a unique product called "The Snuggly" and would like a program to help process customer orders. The program requires the user to enter the customer's name (must be entered), street address, city, province (must be valid) postal code, phone number credit card number, along with the number of Snugglys they wish to purchase (between 1 and 20), and the method of payment as Credit Card or Pay Later (must be C or P). Add a calculation or two and some output. Allow the program to repeat using the method of your choice.

Validations can become a bit more complex when you want to look at allowing only certain characters for an input. For example a name can include upper and lowercase letters along with an ‘ or a -, especially in a last name like "O'Connor-Smith". Just to spice it up I recently saw a last name of Lee6 – so numbers may be included!!

This is set up using an allowable set of characters and testing to ensure the values are in the set. The first is a customer last name, the second is a phone number.

Many times, names are entered as separate inputs for the first name and last name.

```
allowed_char = set("ABCDEFGHIJKLMONPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz-'")
# if the value allows spaces, I usually add it between the upper and lowercase characters.

while True:

        # Since this is a string value the Try/Except is not necessary
```

```
        CustLastName = input("Customer Last Name: ")

        # Now check the name and make sure it is valid.
        if CustLastName == "":
                print("Customer Name must not be blank. Please re-enter.")
        elif set(CustLastName).issubset(allowed_char) == False:
                print("Customer Name contains invalid characters. Please re-enter.")
        else:
                break

while True:

        PhoneNum = input("Enter the customer phone number (999999999): ")

        if PhoneNum == "":
                print("Phone number cannot be blank - Please re-enter.")
        elif len(PhoneNum) != 10:
                print("Phone number must contain 10 digits - Please re-enter.")
        elif PhoneNum.isdigit() == False:  # Rather than a set I used isdigit() to check for numbers.
                print("Phone number must contain numbers only - Please re-enter.")
        else:
                break
```

Here is a program that you can use to practice loops and validations – and a bunch of other things that we have learned along the way.

Computers R Us needs a program to evaluate retail staff on a weekly basis and compare their totals at a regional level.

Input the region name (Must be entered), the salesperson first name and last name separately (Must be entered), their sales for the week (Must be a valid number – cannot exceed 30000.00), and the number of hours they worked during the week (Must be between 10 and 60).

Calculate the gross pay for that employee based on the hourly gross pay and a commission - use a rate of $26.00 per hour, and a commission of 1% on sales. If the commission is less than $250.00, subtract the amount under $250.00 from the gross pay.

Determine a status message that reads “Above Average” if the salesperson sales are greater than 20000.00, “OK” if they are between 10000.00 and 20000.00, and “Below Average” otherwise.

Display the results using the following printer spacing chart.

```
          1         2         3         4         5
  123456789012345678901234567890123456789012345678901234567890123456789012345678
1  Computers R US – Regional Sales Analysis
2
3  Salesperson name: XXXXXXXXXXXXXXXXX Region: XXXXXXXXXXXX
4
5  Salesperson sales: $9,999.99     Hourly pay:   $9,999.99
6                                   Commission:     $999.99
7                                   ----------------------
8  Status:        XXXXXXXXXXXXX     Gross pay:    $9,999.99
```

Set up a loop to process increases in sales from 2 to 20% in increments of 2%. Recalculate the commission based on the new percentages and the new gross pay.

Display the table with the sales increases at the end of the printout starting on line 10 as shown below.

```
9
10 Sales increases from 2 to 20%:
11
12      Increase      Commission       Gross Pay
13      ---------------------------------------
14         ##%         $9,999.99       $9,999.99
15                         :
16         ##%         $9,999.99       $9,999.99
17
18 Do you want to process another employee (Y/N): X
19
```

See you at 1.