# Introduction to Programming

# Functions

- For Loop: A Quick Review
- Indefinite Loops
- Common Loop Patterns
  - Interactive Loops
  - Senteniel Loops
  - **Error Handling**
  - **File Loops**
  - **Nested Loops**
- Computing with Boolean
- Other Common Structures

# Error Handling

```
while num < 0:
  print ('number must be positive. try again')
  num = int (input ('enter a positive number: '))
```

- We can use a while loop similar to this loop to handle input errors
- We can use a try/except block nested within a while loop to handle input errors
- Example

# File Loops

- The biggest disadvantage of our program at this point is that they are interactive.
- What happens if you make a typo on number 43 out of 50?
- A better solution for large data sets is to read the data from a file.
- Example with for loop.
- Can we use `while` loop to read a file?

# File Loop

- We could use `readline` in a loop to get the next line of the file.
- At the end of the file, `readline` returns an empty string, “”
- Example
- Does this code correctly handle the case where there’s a blank line in the file?
- Yes. An empty line actually ends with the newline character, and `readline` includes the newline. “\n” != ““”

# Nested Loops

- In the last chapter we saw how we could nest `if` statements. We can also nest loops.
- Example
- Suppose we change our specification to allow any number of numbers on a line in the file (separated by commas), rather than one per line.

# Nested Loops

- At the top level, we will use a file-processing loop that computes a running sum and count.

```
sum = 0.0
count = 0
line = infile.readline()
while line != "":
    #update sum and count for values in line
    line = infile.readline()
print("\nThe average of the numbers is",
sum/count)
```

# Nested Loops

- In the next level in we need to update the `sum` and `count` in the body of the loop.
- Since each line of the file contains one or more numbers separated by commas, we can split the string into substrings, each of which represents a number.
- Then we need to loop through the substrings, convert each to a number, and add it to `sum`.
- We also need to update `count`.

# Nested Loops

- The loop that processes the numbers in each line is indented inside of the file processing loop.
- The outer while loop iterates once for each line of the file.
- For each iteration of the outer loop, the inner for loop iterates as many times as there are numbers on the line.
- When the inner loop finishes, the next line of the file is read, and this process begins again.

# Nested Loops

- Designing nested loops –
  - Design the outer loop without worrying about what goes inside
  - Design what goes inside, ignoring the outer loop.
  - Put the pieces together, preserving the nesting.

# Class Work

1) A positive number n > 2 is prime if no number between 2 and √n (inclusively) evenly divides n. Write a python program that will ask the user for a positive number x (use while loop and try/except block for error handling) and displays all prime number between 2 and x. Hint: You will have to use nested loop.