Loops continued
Lecture 16

Week 8

looping through a string

A for loop can iterate* through strings

*iterate means to journey through a multi-value object one value at a time

Example: string encryption

Write code to partially encrypt a user's user input by replacing some characters with alternates

- Ask the user to enter a string, use input validation to ensure that the string greater than 3 character long
- Print out an encrypted version of the user's input
  - replace any vowel with *
  - replace any spaces with @
  - replace any digits with X
  - print all other characters as normal

Example: calculate average

Write code to calculate the average of all integers between 5 and 21.

- Write code that generate the sequence of numbers between 5 and 21
- Create a variable to count how many numbers are in the sequence
- Create a variable to sum up all the values in the sequence
- Calculate and print the average

Example: calculate minimum

Write code that asks a user to enter 5 integers and find the lowest number

- Write a loop that will iterate 5 times
- On each iteration ask the user to enter an integer
- Use a try/except structure to ensure that the code will not crash
- Create a variable to store the current lowest number (what will the initial value be?)
- Check if the current number is smaller than the previous lowest

Example: find the longest word

Write code that asks a user to enter words and find the longest word

- Write a loop that iterates until the user wants to stop (stop when the user enters the string “stop”)
- Create a variable to store the length of the longest word and another to store the longest word
- Check if the length of the current word is longer than the previous longest word
- If the word is the new longest word update the longest length variable and the longest word variable