Week 3 (Chapter 1)

X=2 < assignment statement

x=x+2<assignment with expression

print(x)< print function

x is a variable, = & + are operators, 2 is constant and print is a function

-Most code scripts are too long to be in Interactive Python, so we put them in files and tell Python the run the commands there

Iteration variables: variables that change each time theyre put through a loop

print(“Hello world!”)> Hello world!

Week 4 (Chapter 2)

Constants are fixed variables like numbers or letters inside quotes

Reserved words have fixed actions and cant be used as variables>false,true,and,as,assert,break,class,if,def,del,elif,else,ecept,return,for,from.global,try,import,etc.

Variable: a named place in memory where a programmer can store data

Good: spam eggs spam23 \_speed

Bad: 23spam #sign ver.12

Different: spam Spam SPAM

A=12

B=10

Print(A\*B)

=120

Assignment Statement: Consists of expression on the right-hand side and a variable to store the result

How to add decimal places > format(variable,’.2f’)( the .2 is the amount of spaces)

Format is a function, similar to print. Allows you to format variables.

Float and int are data types. Float gives you a fixed decimal while int is a whole number.

Week 5 (Chapter 3)

“If” and “Else” are conditional statements

Indentation is important and so is using a :

Boolean expressions: ask a question and produce a Yes or No result

<Less than <= Less than or equal to ==equal to != not equal

Tabs and spaces may or may not be equivalent

Drawing boxes around code helps to keep organized

Elif is another conditional statement>

If blank:

print(blank)

Elif blank:

print(blank)

Else blank:

print(blank)

Keep intervals straight in ranges or it wont work.

Week 7 (Chapter 5)

Loops have **iteration variables** that change each time through a loop.

n=5 Have to initiate variable.

While n>0: Have to set condition

print(n) prints variable every time it goes through loop

n=n-1 iteration variable, if not it wouldn’t know what to do after entering 5

print(‘blastoff!’) only prints once loop is completed, not in indentation

print(n) will print final variable of end after loop is completed since not in indentation

LOOP THAT WILL NOT WORK

n=5

While n>0:

Print(‘Lather’)

Print(‘Rinse”)

Print(‘Dry off’)

NO ITERATION VARIABLE

#Prof said breaks was bad practice so I didn’t include notes on breaks and continues.

Definite loop

for i in (5,4,3,2,1): green is iteration variable that iterates through the sequence

print(i) the purple is the (body) of code that is executed once for each value in the sequence

print(‘Blastoff!’)

5

4

3

2

1

Blastoff!

friends=(‘Joseph’,’Glenn’,’Sally’)

for friend in friends:

print(‘Happy New Year:’,friend)

print(‘Done!’)

Happy New Year: Joseph

Happy New Year: Glenn

Happy New Year: Sally

print(‘Before’)

for thing in(9,41,12,3,74,15)

print(thing)

print(‘After’)

before

9

41

12

3

74

15

After

Notice how Before and After are not in loop

Average in Loop

Count=0

Sum=0

Print(‘Before’,count,sum)

For value in (9,41,12,3,74,15):

Count=count+1

Sum=sum+value

Print(count,suum,value)

Print(‘After’,count,sum,sum/count)

Before 00

9 9

50 41

62 12

65 3

139 74

154 15

After 154 25

Week 6 (Chapter 4)

-Functions start with the keyword def, followed by the name and then end with :

-Def defines the function, make sure to indent bondy of function.

-Functions in a program don’t actually run unless they’re “Called” by name

-Function examples: print() float()input()type()int()string()

Input() always gives back a string

Argument: Value we pass into function as its input when we call the function

-used to direct function to do different types of work

-Using global variables is BAD practice, instead use parameters

-Parameter: a variable which we use in the function def, a handle that allows the function to access variables outside the function.

Return values return a value to be used as the value of a function call in the calling expression.

Def greet()

Return “hello”

Print(greet(),”glenn”)  
print(greet(),”Sally”)

Hello Glenn

Hello Sally

Fruitful functions produce a result/return value.

-We can use more than one parameter, we got to add more arguments, match number and order of arguments and parameters.

Chapter 6 strings:

String: a sequence of characters

-uses quotes

-+ sign concatenates or brings together

-When a string contains numbers, it is still a string, until converted into number by int()

-You can search inside strings with square brackets=>

Fruit=’banana’

Letter=fruit[1]

Print(letter)

A

X=3

W=fruit(x-1)

Print(w)

N

Len() is a function that takes input parameters(counts letters)

Iteration variable in for loop ‘iterates’ through the string and the body

The “in” operator is checking for one string in another string, returns true or false

String library:

Str.capitalize() str.center(width[,fillchar])

Str.center(width[,fillchar])

Str.endswith(suffix[,start[,end]])

Str.find(sub,start[,end]])

Str.lstrip([chars]) (whitespace)

Str.replace(old,new[,count])

Str.lower()

String.rstrip([chars]) (whitespace)

Str.strip([chars]) (Both beginning and ending whitespace)

Str.upper()

Search and replace can be done through “replace” function, replace(Variable that needs to be replaced, what you want it replaced with)

Chapter 7 Files:

-A text file can be thought of as a sequence of lines

-Before accessing file one would need to use open() function to let python know what variable we’re going to work with, and what we will be doing with the file. The function returns a “file handle” which is a variable used to perform functions on a file.

Handle=open(filename,mode)

Fhand=open(‘mbox.txt’,’r’)

Handle=returns a handle use to manipulate the file

Filename= is a string

Mode= optional and should be ‘r’ if user wants to read file and ‘w’ if user wants to write to the file.

-‘newline” character is used to indicate when a line ends, represented as “\n” in strings. A textfile has newlines at the end of each line.

-a file handle can be a sequence of strings and can use the for statement to iterate through a sequence(ordered set)

Xfile=open(‘mbox.text’)

For cheese in xfile:

Print(cheese)

-We can put an if statement in our for loop to only print lines that meet some criteria.

-Can skip line by using the ‘continue’ statement