Aaron Eberly

CNIT 15502

Week 7 Deliverable

Recursive Functions

A recursive function is a function that calls itself. You CAN call a function within itself. However, this is bad practice, and should not be used often. Anything that can be accomplished through recursion can also happen with a loop.

def callMeBack(n):

    print(n)

    if n > 0:

        callMeBack(n - 1)

callMeBack(4)

The above code is recursive, since it calls callMeBack() from within the function callMeBack(). It will print numbers backwards from 4 – 0. However, observe how this could instead be done with a for loop:

def callMeBack(n):

    for num in range(n, -1, -1):

        print(n)

        n -= 1

callMeBack(4)

Look at the following code. By having a return statement that calls itself, it adds up all the values from 1 to n:

def summer(n):

    if(n == 1):

        return 1

    else:

        return(n + summer(n - 1))

returnVal = summer(5)

print(returnVal)

Let’s walk through the code:

summer(5) returns 5 + summer(4)

summer(4) returns 4 + summer(3)

summer(3) returns 3 + summer(2)

summer(2) returns 2 + summer(1)

summer(1) returns 1

So in total, summer 5 returns 5 + 4 + 3 + 2 + 1

All of these are added to return 15.

Attached with this document will be the code for the City Builder assignment, with many in-line comments to explain the code step by step.