**Assigned: Wednesday, October 5 David Schmidt**

**Due: Thursday October 12th at classtime**

**Objective: Working with functions to modularize solutions with a “divide and conquer” approach.**

A. Do all of the “Let’s Try It” exercises in Chapter 5. Obtain the signature of a classmate or your teacher when you SHOW them the results and the signer VERIFIES completion (5 pts)

1. Lance Molz 6. Jax Milne

2. Garret Fancher 7. Tanner Allen

3. Christian Trout 8. Tanner Allen

4. Tyler Smith 9. Tanner Allen

5. Dr. DeClue

B. Answer Chapter Exercise Questions 1, 3 and 5 on pages 202-203 in a MS Word Doc. Retype the question before answering it. **Print this document** **and staple it to this sheet to hand in on the due date**. (5 pts)

C. Write one Python program containing the requested functions in Python Programming Exercises P1, P3 and P5. Test your functions by calling them with some sample parameters. Use print statements to display the results on the console. When the displayed results show the functions work correctly, collect a screen capture of the output, paste it into a Word doc. Copy your code into this document as well and **print this document and staple it to this sheet to hand in on the due date.** (10 pts)

D. Write an Earsketch song which uses at least three functions to define a song. The first function should define an intro part of the song, the second function should define the main part of the song, and the third function should define an outro for the song. Pass parameters to the functions which define the starting and stopping measures for the sections. **Paste the link for your song into a Word doc and submit it to the dropbox for this assignment**. (5 pts)

P1. def zeroCheck(n1,n2,n3):

if(n1==0) or (n2==0) or (n3==0):

return True

else:

return False

print(zeroCheck(0,1,2))

print(zeroCheck(0,3,4))

print(zeroCheck(5,6,7))

P3.

def modCount(n,m):

i = 1

count = 0

if m <= n:

while i <= n:

if i % m == 0:

count += 1

i += 1

return count

print(modCount(10,5))

print(modCount(18,6))

print(modCount(20,7))

P5. def printAsterisks(n):

if (n > 75):

print('\*' \* (75))

else:

print('\*' \* (n))

printAsterisks(25)

printAsterisks(50)

printAsterisks(100)