**1) For the recur method in the SimpleRecursion class, answer the following:**

**a. What is the output of version 1 called with argument 0.**

Entering recur with k = 0

Recursing with k = 0

Entering recur with k = 1

Recursing with k = 1

Entering recur with k = 2

Recursing with k = 2

Entering recur with k = 3

Recursing with k = 3

Entering recur with k = 4

Leaving recur with k = 4

Leaving recur with k = 3

Leaving recur with k = 2

Leaving recur with k = 1

Leaving recur with k = 0

**b. What is the output of version 2 called with argument 0.**

count = 0

count = 1

count = 2

count = 3

count = 4

count = 5

**c. What is the output of version 3 called with argument 0.**

There is no output, it only works if the argument is equal to 5.

**d. What is the output of version 4 called with argument 0.**

count = 0

count = 1

count = 2

count = 3

count = 4

**e. What is the output of version 5 called with argument 0.**

count = 0

**f. What is the output of version 6 called with argument 0. How can we run all 6 versions at once? (hint: name overloading does not work)**

count = 0

count = 1

count = 2

count = 3

count = 4

We can run all methods at once if you just change the name of the methods and call them with the appropriate argument.

**2) After filling-in the reverse6 method in the NaiveReverse class, can you think of how to extend this algorithm for reverse7 method? Explain.**

This can be easily extended to reverse 7 by just following the same pattern. All one needs to do is to:

public String reverse7(String s){

return(s.substring(6,7)+s.substring(5,6)+s.substring(4,5)+

s.substring(3,4)+s.substring(2,3)+s.substring(1,2)+s.substring(0,1));

}

**3) Answer the following questions concerning the reverse9 method given in Task 2:**

**a. A student tests the reverse9 method on the String "confident". What will it return?**

It will return:

ctnedifno

**b. What should be the proper reversed string for "confident"?**

It should be:

tnedifnoc

**c. Suggest how to fix the code so the method does it correctly.**

We can fix the code by modifying it to be:

public String reverse9 (String s) {

String partial = reverse8(s.substring(0, 8));

String firstLetter = s.substring(8, 9);

return (firstLetter + partial);

}

**4) Answer the following questions concerning the digitCount method in the DigitCount class:**

**a. What is returned from the call digitCount(0)?**

The call for digitCount(0) returns 1.

**b. What is returned from the call digitCount(10)?**

The call for digitCount(10) returns 2.

**c. What is returned from the call digitCount(314159)?**

The call for digitCount(314159) returns 6.

**d. Should the digit count increase if the value is 0? If not how should the code change**

The digit count for zero should not increase for zero, it should remain at 1. The code does not change much it looks as follows:

public static int digitCount(int value) {

if(value == 0)

return 1;

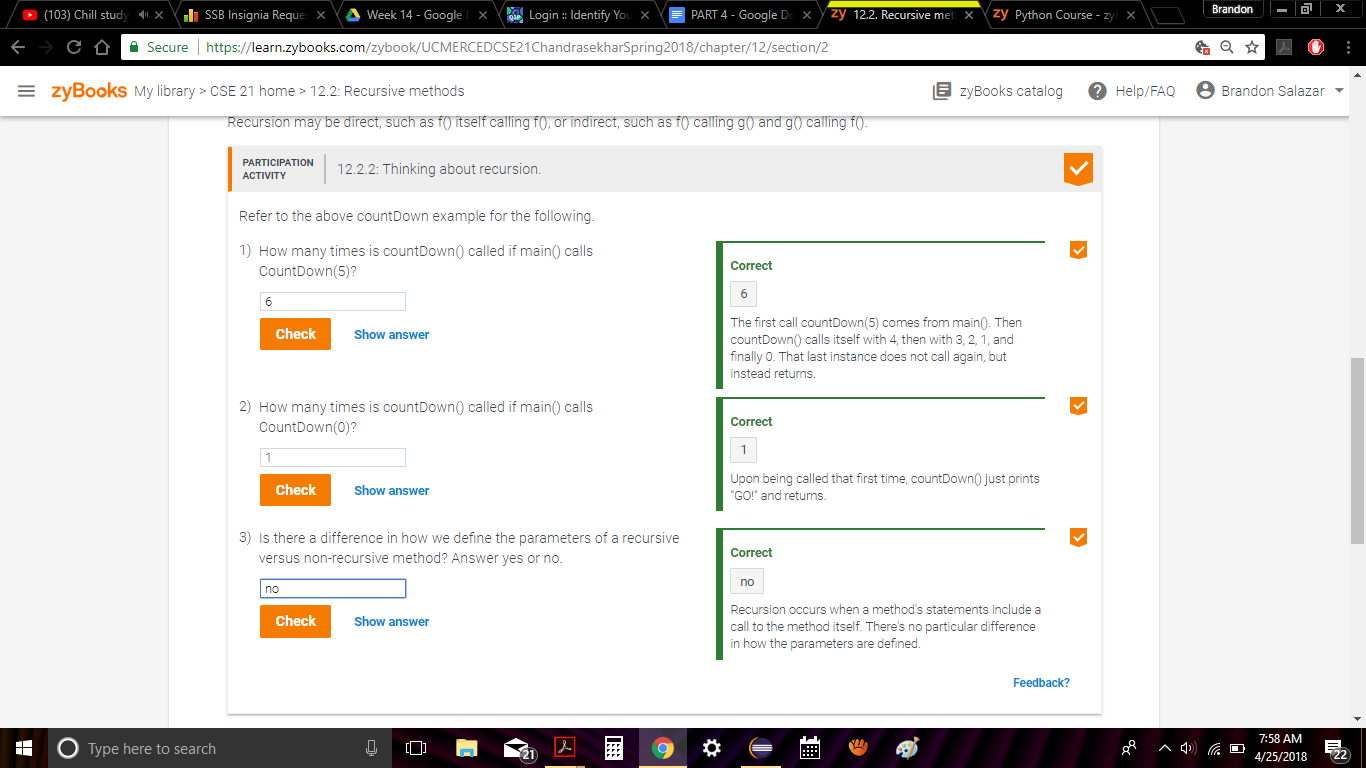
else

return 1 + digitCount(value/10)

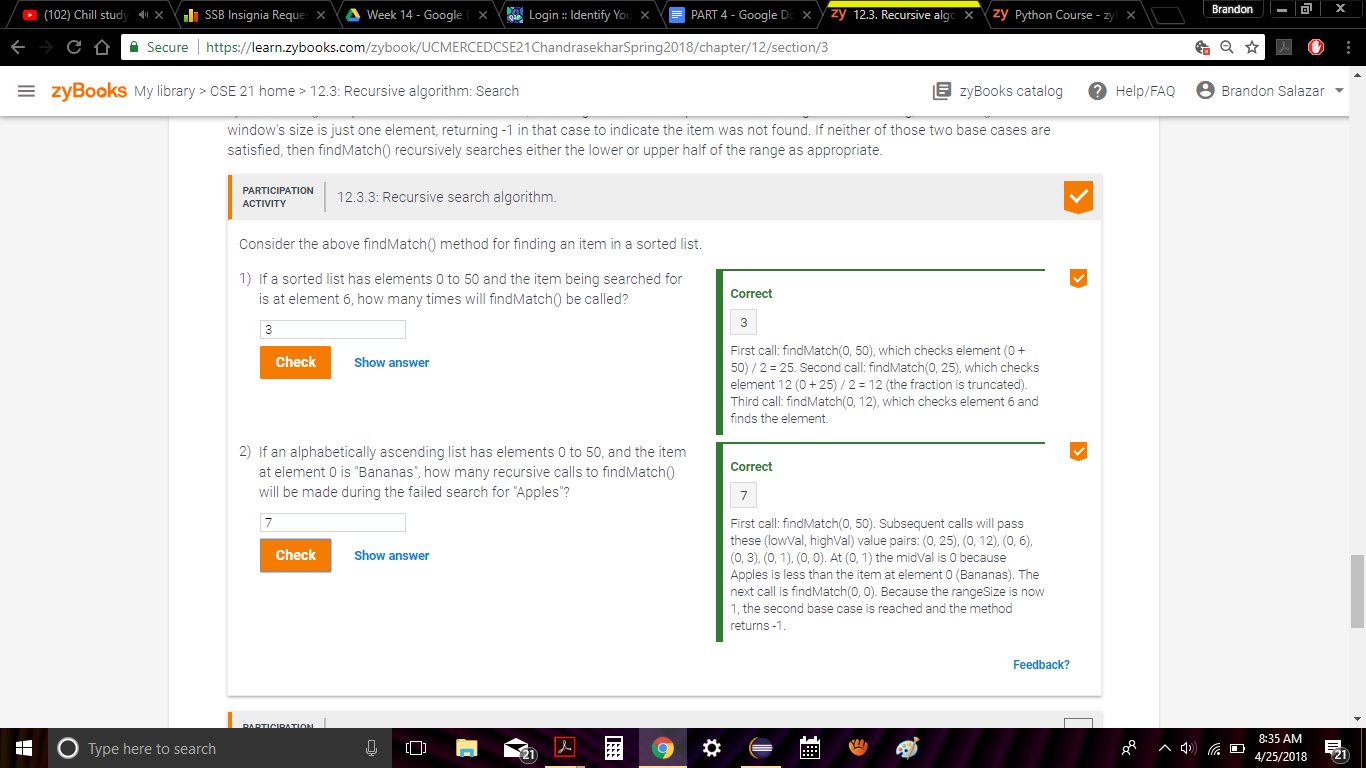
}

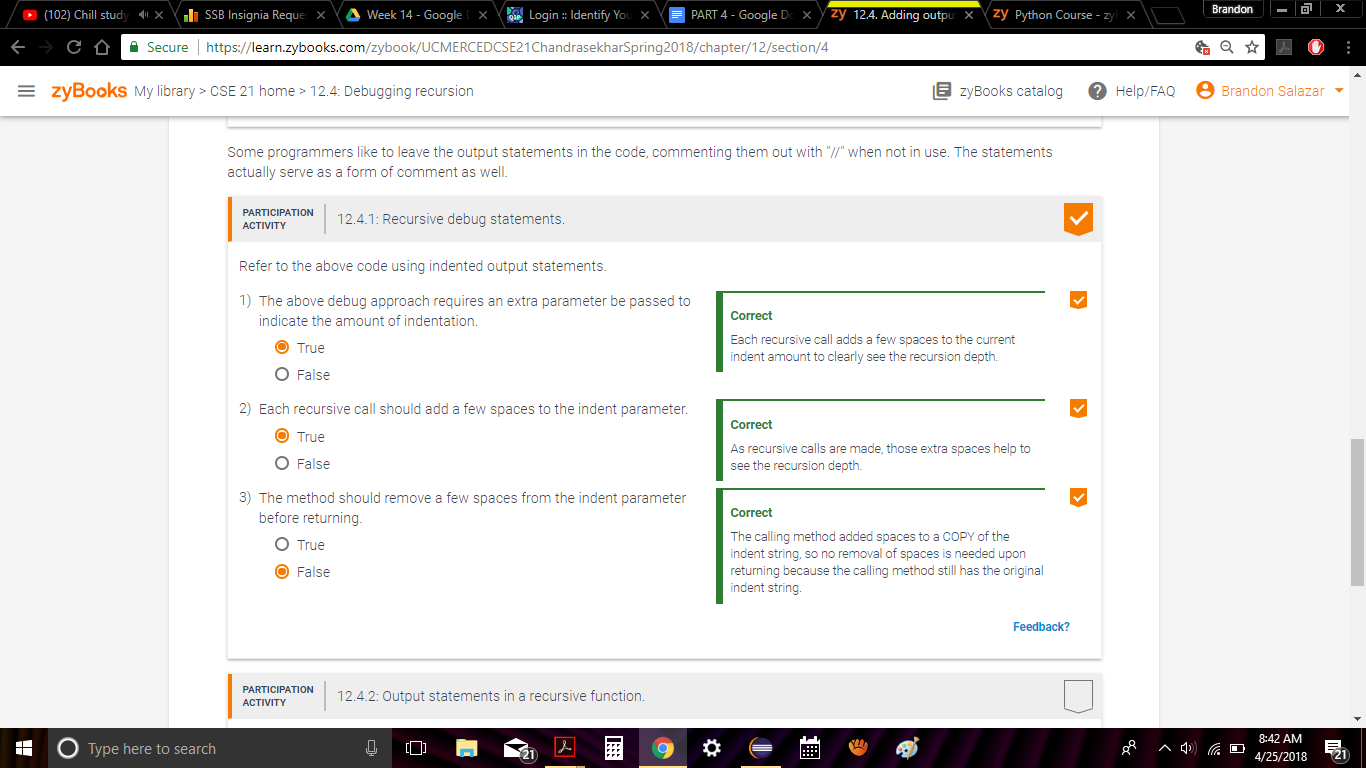
**e. What are other values of value for which digit count should not increase? Change the code to include those values.**

It should not increase for any of the first 10 values(0 - 9) which are all 1 digit in total.

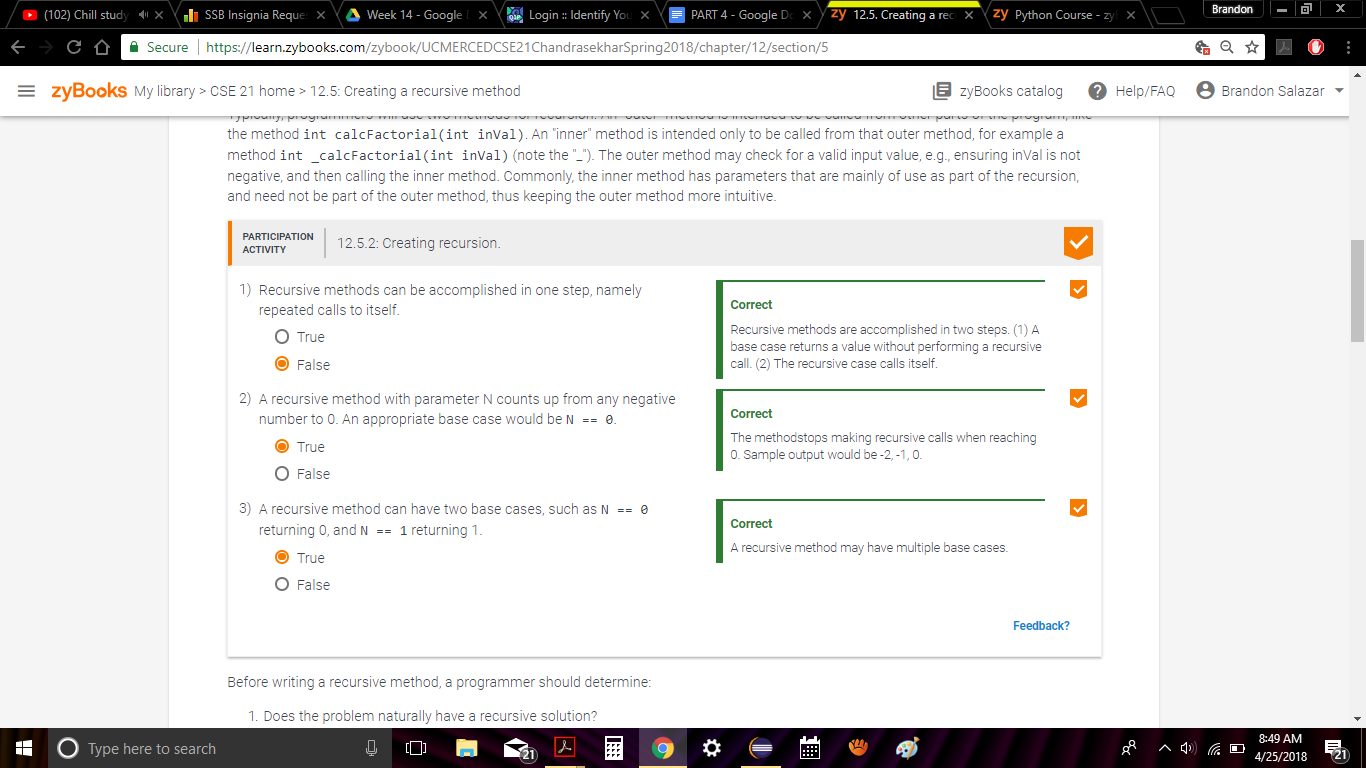
**12.2.2**

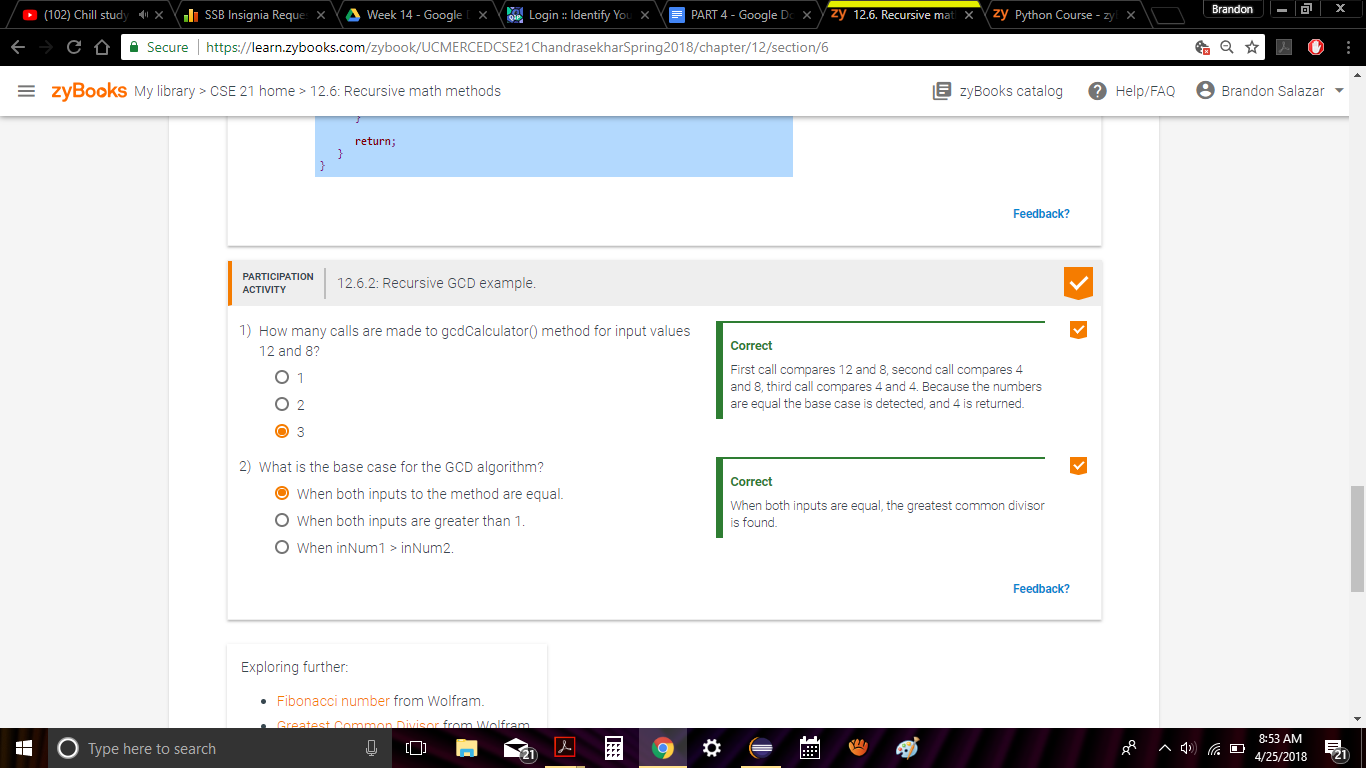
**12.3.3**

****

**12.4.1**

**12.5.2**

****

**12.6.2**