CSCI 150
Exam 1 Solutions

1. What will the following program print?

```
def C(n):
    for j in range(0, n):
        print(j, end="  ")
    print()

def B( n ):
    return 2*n

def A(n):
    x = B(n)
    C(x)
    C(B(x))

def main():
    A(2)

main()
```

When we call A(2) it sets x=B(2) = 4 and then calls C(4) and C(B(4)), which is C(8). C(n) prints the numbers from 0 to n-1 on one line. Altogether this prints

0 1 2 3
0 1 2 3 4 5 6 7

2. What will this program print? Read it carefully.

```
def change(x):
    x = 3

def main():
    for x in range(1, 6):
        print(x)
        change(x)
         if x == 3:
            done = True

main()
```

change(x) does nothing at all since change's variable x is different from main's x. Also, setting done=True does nothing; it certainly doesn't break you out of the for loop. Remember that Python doesn't know English. Setting done=True is no different than setting bob=True or flibbertygibbet=True; the only impact regards what you do with that variable, and this program does nothing with variable done. main( ) might as well just be

```
for x in range(1, 6):
    print x
```

and this prints

```
1
2
3
4
5
```

3. In the following program I get an error on the print statement:
   print(“That string has %d spaces.”%num)

. The error message says
   NameError: name 'num' is not defined
Fix the program so that it correctly prints the number of spaces in any string we enter.

```
def CountSpaces(s):
   # This returns the number of space characters in string s.

        num = 0
        for letter in s:
                if letter == " ":
                        num = num+1
        return num

def main():
        s = input( "Enter a string: " )
        CountSpaces(s)
        print("That string has %d spaces." %num )

main()
```

The error message says that variable num is not defined in main. Function main( ) can’t see the variable num in CountSpaces. The real problem is that we aren’t making use of the value returned by CountSpaces. Change main( ) to

```
def main():
        s = input( "Enter a string: " )
        n = CountSpaces(s)
        print("That string has %d spaces." %n )
```

If you prefer you can use any other variable name in place of n, including num.

4. The following program is supposed to read numbers from the user and print the corresponding factorial. It mostly works – if I give it 3 it prints 6; if I give it 5 it prints 120, if I give it 10 it prints 3628800. When I give the input 0 the program is supposed to print 1 as the result of factorial(0) and then exit. However, it doesn't exit or print anything on input 0; it just sits there caught in some kind of loop. Explain in one or two English sentences what is wrong. I'm not asking you to fix or rewrite the program, just explain why it doesn't work for input 0.

```
def factorial(n):
    total = 1
    x = n
    while x != 1:
        total = total*x
        x = x-1
    return total

def main():
    done = False
    while not done:
        x = input( "Enter a number or 0 to exit: " )
        if x == 0:
            done = True
        print factorial(x)

main()
```

There is nothing wrong with this program calling print(factorial(0)) (other than that I forgot the parentheses for print); the problem is that factorial(0) never returns a value. When function factorial gets argument 0 it sets x=0, and then starts subtracting values from x, waiting for it to become 1. Of course, if x starts out less than 1, subtraction will never turn it into 1.

5. Write the Rumplestiltskin program. This repeatedly asks the user "What is my name?" If the user replies "Rumplestiltskin" the program prints "Drat!" and halts. If the user replies anything but "Rumplestiltskin" the program says "NO!" and asks again "What is my name?" Here is a sample run, with what the computer prints in **boldface**:

What is my name? bob
NO!
What is my name? Frodo
NO!
What is my name? Some annoying character from a German folktale
NO!
What is my name? Rumplestiltskin
Drat!

```
def main( ):
        done = False
        while not done:
                name = input( "What is my name? " )
                if name == "Rumplestiltskin":
                        done = True
                        print( "Drat!" )
                else:
                        print( "NO!" )
```

The most common mistake here was to put the input statement before the while loop. If that statement isn't in the loop you only get input once. This is the point of using that formulaic while loop with variable done; it allows you to delay deciding when to quit the loop until you are inside the loop body.

6. Write a method **Triangle(n)** that draws a triangle made of *'s, where n is the number of rows of the triangle. For example, when n is 2 this should draw

```
*
**
```

When n is 4 it should draw

```
*
**
*  *
****
```

When n is 6 it should draw

```
*
**
*  *
*    *
*      *
******
```

```
def Triangle(n):
        print("*")   # print the top line
        for i in range(0, n-2):  # line i has a star, i spaces and a second star
                print( "^", end="")
                for j in range(0, i):
                        print( " " , end = "")
                print( "*" )
        for j in range(0, n):  # the bottom line has n stars
                print( "*", end="")
        print()
```

Alternatively, if you like multiplying strings:

```
def Triange(n):
        print( "*" )
        for i in rnage(0, n-2)"
                print( "*" + " "*i + "*" )
        print( "*"*n )
```

7. Write a function **singles(L)** whose argument L is a list of strings. The function should print every string that appears in L only once. For example,
singles([“Bob”, “Harry”, “Hermione”, “Ron”, “Neville”, “Bob”, “Harry”, “Ron”])
prints

```
Hermione
Neville
```

One way to do this is to count the number of times you see each element, and print the ones whose count is one:

```
def countInList(x, L):
    count = 0
    for item in L:
        if item == x:
            count = count + 1
    return count

def singles(L):
    for x in L:
        if countInList(x, L) == 1:
            print(x)
```

Another way is to use lists to keep track of what you have seen so far:

```
def singles(L):
    seenSoFar = [ ]
    seenTwice = [ ]
    for x in L:
        if x in seenSoFar:
            seenTwice.append(x)
        else:
            seenSoFar.append(x)
    for x in L:
        if x not in seenTwice:
            print(x)
```