Name_____

CSCI 150
Final Exam Soluutions
May 12, 2016

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

```
def F(x):
    total = 0
    for i in range(1, x):
        total = total + i
    print( G(total) )

def G(x):
    return x*x

def main():
    L = [2, 3, 1]
    for x in L:
        F(x+1)

main()
```

This prints

9
36
1

2. What will this print? Again, read it carefully.

```
def F(x):
    if x == 1:
        return x
    elif x%2 == 0:
        return G(x)
    else:
        return F(x-2)

def G(x):
    if x == 0:
        return x
    elif x%2 == 0:
        return G(x-2)
    else:
        return F(x)

def main():
   for i in range(0, 6):
       print(i, F(i))

main()
```

This prints

0 0
1 1
2 0
3 1
4 0
5 1

3. The following program tries to give its own function for changing the contents of a list. The call

change(L, “add”, 5)

should append the value 5 onto list L; the call

change(L, “remove”, 5)

should remove all instances of value 5 from L.

```
def change( L, operation, value):
    if operation == "add":
        L.append(value)
    elif operation == "remove":
        L1 = []
        for x in L:
            if x != value:
                L1.append(x)
        L = L1

def main():
    myList = []
    change( myList, "add", 3 )
    change( myList, "add", 5 )
    change( myList, "add", 3 )
    change( myList, "remove", 3 )
    change( myList, "add", 7 )
    change( myList, "remove", 5 )
    change( myList, "add", 9 )
    print( myList )

main()
```

a) What will this program print?

It prints [3,5,3,7,9]

b) Explain your answer to (a). One sentence should be sufficient.

The deletes create a new list and work with that, so they have no impact on myList.

4. When I run the following program it incorrectly says

   The string ‘bob’ does not contain a vowel.

   Fix this program so that the HasVowel(s) function correctly determines whether string s contains a vowel. Note that I’m not asking you to explain the mistake; just say how to fix it. You can write directly over my code.

```
def IsVowel(x):
    if x in "aeiou":
        return True
    else:
        return False

def HasVowel(s):
    for letter in s:
        if IsVowel(letter):
            return True
        else:
            return False
def main():
    if HasVowel( "bob" ):
        print( "The string 'bob' contains a vowel." )
    else:
        print( "The string 'bob' does not contain a vowel." )

main()
```

Change to:

```
for letter in s:
    if IsVowel(letter):
        return True
return False
```

5. I want a program that will give the user at most 5 guesses of my name. If they don't get it right in 5 guesses the program should halt. Here is an attempt at writing that program:

```
def main():
    done = False
    while not done:
        for turn in range(0, 5):
            guess = input( "What is my name? " )
            if guess == "bob":
                print( “Yess!!!” )
                done =  True

main()
```

a) What will this program do if I guess "bob" on the 3rd guess?

It will ask for 2 more guesses.

b) What will this program do if I guess incorrectly 5 times?

It will ask for 5 more guesses.

c) Write a correct program that will give the user at most 5 guesses of my name.

```
def main():
    done = False
    count = 0
    while not done and count < 5:
        guess = input( "What is my name? " )
        count = count + 1
        if guess == "bob":
            print( “Yess!!!” )
            done =  True

main()
```

6. The following program tries to build a class structure that will contain animals and the sounds they make. The lines

```
c = Cow( )
print(c)
```

in main( ) should produce the output

```
The cow says ‘moo’.
```

Unfortunately, this gets an error message: 'Cow' object has no attribute 'species'.

```
class Animal:
    def __init__(self, species):
        self.species = species
        self.voice = ""

    def __str__(self):
        return "The %s says '%s'"%(self.species, self.voice)

class Cow(Animal):
    def __init__(self):
        Animal.__init__(self, “cow”)
        self.voice = "moo"

def main():
     c = Cow()
     print(c)

main()
```

a) Explain in English what the error message means.

The Cow class didn’t run the Animal constructor, so the variable self.species was never created.

b) Modify the code so this will print: The cow says ‘moo’. You can write over my code.

See the line in red in the code.

T

7. **Write a function sumFactors(n) that returns the sum of the positive factors of number n** (i.e., all of the numbers that evenly divide into n), **including 1 and n**. For example, sumFactors(10) is 18 because 1+2+5+10 is 18. Similarly, sumFactors(12) is 28 because 1+2+3+4+6+12 is 28 and sumFactors(6) is 12 because 1+2+3+6 is 12. Remember that we can tell if number d divides evenly into n by checking if n%d is 0.

```
def sumFactors(n):
        sum = 0
        for i in range(0, n+1):
                if n%i == 0:
                        sum == sum+i
        return sum
```

8. File “words.txt” contains a few thousand words, one word per line. I am interested in words with duplicated letters: “elephant” has a duplicated ‘e’, “bumblebee” has both ‘b’ and ‘e’ duplicated. **Write a program that reads file “words.txt” and prints the letter that is duplicated in the most words.** If there is a tie you can print any one of the most frequently duplicated letters. For example, if the only words in the file were “elephant”, “bumblebee”, and “corkscrew” the winner would be ‘e’, since ‘e’ is duplicated in 2 of the words, while ‘b’, ‘c’, and ‘r’ are duplicated in 1.

```
def isDuplicated(letter, s):
        count = 0
        for c in s:
                if c == letter:
                        count = count +1
                        if count == 2:
                                return True
        return False

def main():
        Counts = { }
        F = open(“words.txt”, “r”)
        for line in F:
                line = line.strip( )
                for letter in line:
                        if isDuplicated(letter, line):
                                if letter in Counts.keys( ):
                                        Counts[letter] = Counts[letter]+1
                                else:
                                        Counts[letter] = 1
        max = 0
        bigLetter = “”
        for letter in Counts.keys():
                if Count[letter] > max:
                        max = 0
                        bigLetter = letter’
        print( bigLetter)
```

9. **Write a recursive function isSorted(L)** that returns True if list L is ordered from smallest to largest and False otherwise. For example, isSorted([1,2,2,3,5,6,6,6,7]) should return True, while isSorted([1,2,2,3,2,5,6]) should return False. If you don't see how to do this recursively you will get some credit for doing it with a loop.

```
def isSorted(L):
    if len(L) <= 1:
        return True
    elif L[0] > L[1]:
        return False
    else:
        return isSorted( L[1:] )
```

10. For this question you need to **write two classes** that together drill students in addition problems.

class Problem represents a single addition problem:

- The constructor takes no arguments and makes two random numbers between 1 and 100.
- Method Answer( ) returns the sum of the numbers created in the constructor.
- Method Pose( ) prints the problem, gets a response from the user, and returns this response.

For example, if the two numbers are 23 and 45, Pose( ) might print `23 + 45 =`
and then return whatever the user types, which hopefully will be 68, which is what is returned by Answer() .

class Exam gives problems and keeps track of how many of the user's answers were correct.

- The constructor takes as an argument the number of problems to give. Exam(10) should give 10 questions.
- Method GiveQuestion() constructs a Problem, Poses it, and checks the user's answer against the correct answer.
- Method GiveExam() calls GiveQuestion( ) the number of times listed in the constructor.
- Method Report() prints the number of questions asked and the number of correct answers.

You only need to write the classes, but a typical application program might be:

```
def main( ):
        e = Exam(10)
        e.GiveExam( )
        e.Report( )
```

```
class Problem:
    def __init__(self):
        self.A = random.randint(1, 100)
        self.B = random.randint(1,100)
    def Answer( self):
        return self.A+self.B
    def Pose( self ):
        print( "%d+%d= "%(self.A, self.B), end="" )
        ans = input( )
        return int(ans)
class Exam:
    def __init__(self, n):
        self.problemCount = n
        self.correct = 0
    def GiveQuestion(self):
        p = Problem( )
        if p.Pose( ) == p.Answer( ):
            self.correct = self.correct + 1
    def GiveExam( self ):
        for i in range(0, self.problemCount):
            self.GiveQuestion( )
    def Report(self):
        print("This exam gave %d questions."%self.problemCount)
        print("There were %d correct answers."%self.correct )
```