CS 100 Spring 2016 Midterm 1

Monday, Feb 15, 2016

There are 13 questions on this test. Record your answers to the first 10 questions by circling a letter below. Answer questions 11, 12 and 13 on the attached pages. We have also provided scratch pages for writing trial code for the programming problems and tracing code for the multiple choice problems. Work on the scratch pages will not be graded. The value of each question is

1-10 multiple choice (4 points each)
11-13 programming (20 points each)

There is a penalty of one point if your name is not clearly legible and one point if your section is not correct.

Allocate your time accordingly. You may receive partial credit for questions 11, 12 and 13. Answer them as completely as you can. If you finish early, use the extra time to double check your work. When you are done, hand in this answer sheet (including programming question solutions) and sign the exam attendance sheet.

You may use the summary of Python language elements that is provided. You may not use other notes, books or electronic devices. If you have brought a cell phone or other mobile device you must leave it with the proctor during the exam. X

Good luck!

Name (print clearly) _____

Student ID _____ Section (see below) _____

02 TF 10:00 Vaks; **04** MR 2:30 Cohen; **06** MR 2:30 Diallo; **08** WF 10:00 Diallo; **10** MW 10:00 Spirollari; **12** TF 4:00 Vaks; **14** TF 2:30 Vaks; **16** MR 10:00 Diallo; **18** WF 1:00 Diallo; **102** W 6:00 Vaks

Q1 a b c d e

Q2 a b c d e

Q3 a b c d e

Q4 a b c d e

Q5 a b c d e

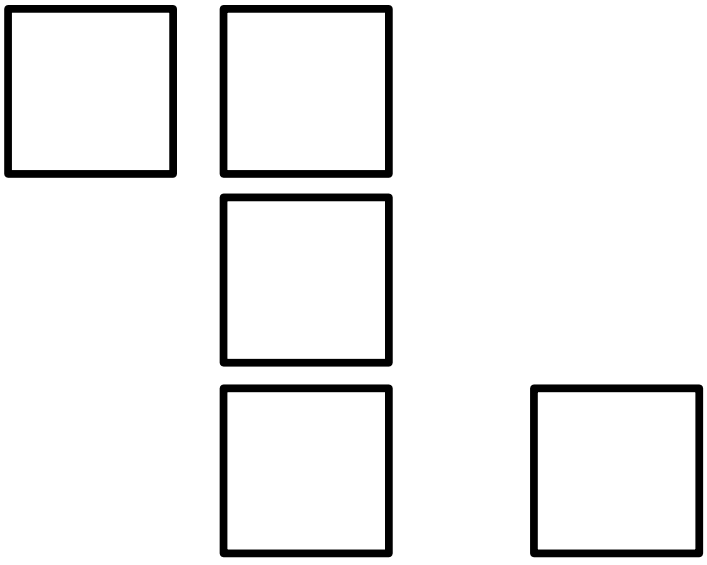
Q6 a b c d e

Q7 a b c d e

Q8 a b c d e

Q9 a b c d e

Q10 a b c d e



Write code for Question 11 here. Use vertical lines for indentation.

Write code for Question 12 here. Use vertical lines for indentation.

Write code for Question 13 here. Use vertical lines for indentation.

Question 1

```
snap_freeze = True
blizzard = False

if snap_freeze and blizzard:
    print('1')
elif not snap_freeze or not blizzard:
    print('2')
if snap_freeze or blizzard:
    print('3')
else:
    print('4')
```

a. 1
b. 2
c. 3
d. 4
e. none of the above

Question 2

```
letters = ['p','y','t','h','o','n']
concat = letters[3] + letters[-2] + letters[2]
print(concat)
```

a. toy
b. hot
c. thy
d. IndexError: negative index
e. none of the above

Question 3

```
someStrings = ['cs', '100', 'midterm']
print(someStrings[1:2])
```

a. 0
b. ['cs', '100']
c. ['100']
d. ['100', 'midterm']
e. none of the above

Question 4

```
aList = [2, -1, -2]
sliceA = aList[:2]
sliceB = aList[-1:]
print(sliceA + sliceB)
```

a. [-1, -2]
b. [2, -1, -2]
c. [2, -2]
d. [2, -1, -2, -2]
e. none of the above

Question 5

```
import turtle
s = turtle.Screen()
t = turtle.Turtle()
for i in range(5):
    if i//4 == 0:
        t.forward(100)
        t.right(90)

# Hint: '//' is integer division
```

a. two adjacent sides of a square
b. three sides of a square
c. a square
d. ZeroDivisionError: division by zero
e. none of the above

Question 6

```
def letterCount(aStr, aLetter):
    if aLetter in aStr:
        return True
    elif aStr.count(aLetter) > 1:
        return 'many'
    return False
result = letterCount("loop", 'o')
print(result)
```

a. True
b. True
   many
c. False
d. SyntaxError: multiple return statements
e. none of the above

Question 7

```
electionYear = True
leapYear = True
if electionYear:
    print('lots of ads')
if not leapYear:
    print('Feb has 28')
else:
    print('quadrennial')
```

a. lots of ads
b. lots of ads
   Feb has 28
c. lots of ads
   quadrennial
d. SyntaxError: multiple 'if'
e. none of the above

Question 8

```
caps = 'ABCDEFGHIJKLMNOPQRSTUVWYXY'
vowels = 'aeiouAEIOU'
etiquette = 'do not SHOUT'
myStr = ''
for letter in etiquette:
    if letter in vowels or letter in caps:
        myStr += letter
print(myStr)
```

a. ooSHOUT
b. ooOSHOUT
c. o o SHOUT
d. ooOSHUT
e. none of the above

Question 9

```
def compareSize(anObject, length):
    objectLen = len(anObject)
    if objectLen == length:
        return 'equal'
    if objectLen == 0:
        return '0'
    else:
        return 'not 0'
    return objectLen
print(compareSize('week', 7))
```

a. equal
b. not 0
c. not 0
   4
d. SyntaxError: multiple return statements
e. none of the above

Question 10

```
def rangeExample(aString, stop, interval):
    returnVal = ''
    myRange = range(1, stop, interval)
    for idx in myRange:
        returnVal += aString[idx]
    return returnVal
print(rangeExample('Do unto others', 13, 4))
```

a. Dno
b. otts
c. ooe
d. IndexError: index out of range
e. none of the above

Question 11A (10 points)

Write a function named *drawSquare*. The function *drawSquare* takes two parameters:

- a turtle, *t*
- an integer, *length*, that is the length of a side of the square.

The function *drawSquare* should use the parameter *t* to draw the square. Do not make any assumptions about the initial up/down state of the turtle, its position on the screen or its orientation. The function *drawSquare* should begin drawing with the turtle at its initial position and orientation. When *drawSquare* returns, the turtle should again be in its initial position and orientation.

You must use a loop for repeated operations.

Question 11B (10 points)

Write a function named *drawSquares* that calls *drawSquare* to draw a specified number of squares.

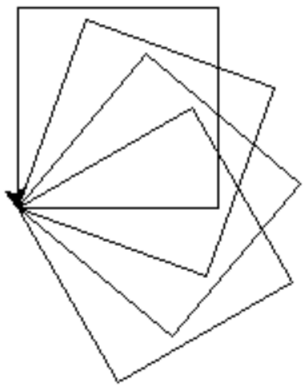
The function *drawSquares* takes four parameters:

- a turtle *t*
- an integer *size*
- an integer *num*, the number of equal signs to draw
- an integer *angle*, the clockwise rotation between successive squares

(Hint: you do not need to import the turtle module or create a screen and turtle. That would have to have been done before *drawSquares* is called.)

For example, the following would be correct output.

```
import turtle
s = turtle.Screen()
snapper = turtle.Turtle()
drawSquares(snapper, 100, 4, 20)
```



Question 12 (20 points)

Write a function named *bigCount* that counts how many elements in a list of numbers are greater than a threshold value and returns that count.

The function *bigCount* takes two parameters:

- *numList*, a list of numbers
- *threshold*, in integer

For example, the following would be correct output:

```
>>> someNums = [-5, 6, 3, 0, 7]
>>> print(bigCount(someNums, 2))
>>> 3
```

Question 13 (20 points)

Write a function named *getDate*. The function *getDate* takes one parameter:

- *message*, a string

In addition to the parameter *message*, the function *getDate* should prompt the user for two items of input: first the month, then the day.

The function *getDate* should produce two kinds of output. It should

- print out the month, day and message, separated by spaces
- return a string consisting of the month and day, separated by a space

For example, the following would be correct input and output:

```
>>> today = getDate('is a great day!')
What month is it? February
What day is it? 15
February 15 is a great day!
>>> print(today)
February 15
```