CS 115, Fall 2017 – Test 3

Name: ________________________________ Date: ___________________

Pledge: ____

Total: ___ / 75

Closed book: no textbook, no electronic devices, one sheet of paper with notes. Read each question carefully before answering, as there is **no partial credit for any response**! You may work out the solution to each question within the test itself, but **only your answers on this cover page will be graded**.

Question 1

a) **[7, 66]** (3 points)

b) **55** (3 points)

c) **[9, 8]** (4 points)

Question 2

B (5 points)

Question 3

a) **5** (5 points)

b) **IK** (5 points)

Question 4

a) **B** (4 points)

b) **is k < n?** (2 points)

c) **is lst[k] > lst[k+1]?** (2 points)

d) **k += 1** (2 points)

Question 5

a) **data** (3 points)

b) **row** (3 points)

c) **val** (3 points)

d) **total** (3 points)

e) **n ** 2** or **(n * n)** (3 points)

Question 6

a) **subclass** (2 points)

b) **inherits** (2 points)

c) **encapsulation** (2 points)

d) **constructor** (2 points)

e) **private** (2 points)

Question 7

a) **BankAccount(456, 130.00)** (2 points)

b) **C overridden** (2 points)

c) **return self.__minimum_balance** (2 points)

d) **@minimum_balance.setter** (2 points)

e) **super().__str__()** (2 points)

Question 8

a) **LightBulb** (2 points)

b) **NeonLight** (2 points)

c) **True** (2 points)

d) **False** (2 points)

e) **True** (2 points)

*** parentheses required for (n * n)**

Question 1 (10 points)

Consider the following code in each section. Assume each section is independent. What is printed on the screen?

```
a) L = [ [9, 8], [7, 6], [5] ]
   M = list(L)
   M[1][1] = 66

   print(L[1])

b) L = [ [9, 8], [7, 6], [5] ]
   M = L
   M[2] = 55

   print(L[2])


c) L = [ [9, 8], [7, 6], [5] ]
   M = deepcopy(L)
   M[0][0] = 99
   M[0][1] = 99

   print(L[0])
```

Question 2 (5 points)
Consider the following code:

```
def binary_search(lst, key):
    low = 0
    high = len(lst) - 1
    while high >= low:
        mid = low + (high - low) // 2
        if key < lst[mid]:
            high = mid - 1
        elif key > lst[mid]:
            low = mid + 1
        else:
            return mid
    return -low – 1
```

This function may fail if it is applied to a list that is not sorted. For which of the following lists `lst` will **binary_search**(lst, 'C') return a negative number? Select one correct answer.

○ A. ['A', 'B', 'C', 'D', 'E', 'F', 'G']
○ B. ['G', 'F', 'E', 'D', 'C', 'B', 'A']
○ C. ['A', 'C', 'D', 'G', 'E', 'B', 'F']
○ D. ['B', 'A', 'D', 'C', 'F', 'E', 'G']
○ E. ['D', 'F', 'B', 'A', 'G', 'C', 'E']

Question 3 (10 points)
Consider the following code in each section. Assume each section is independent. What is printed on the screen?

```
a) try:
       a = int('5')
       print(a)
       sys.exit(0)
   except ValueError:
       print('V')
       sys.exit(1)
   except TypeError:
       print('T')
       sys.exit(1)
```

```
b) lst = [1, 2]
   try:
       a = lst[8 % 3]
       print(a, end='')
   except ValueError:
       print('V', end='')
       sys.exit(1)
   except IndexError:
       print('I', end='')
   print('K', end='')
   sys.exit(0)
```

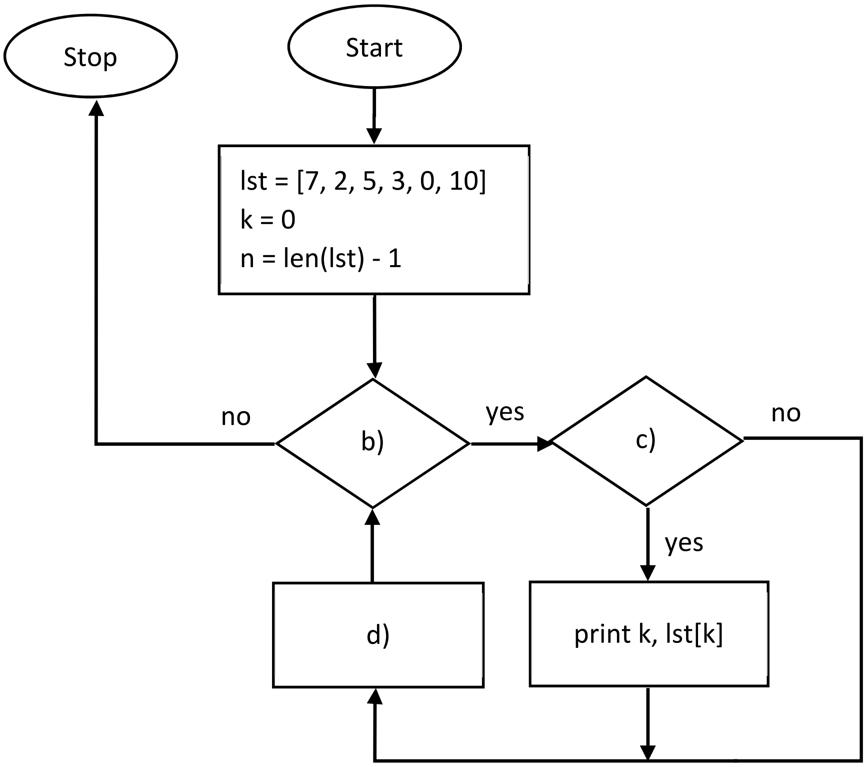
Question 4 (10 points)

Consider the following code:

```
lst = [7, 2, 5, 3, 0, 10]
k = 0
n = len(lst) - 1
while k < n:
    if lst[k] > lst[k + 1]:
        print(k, lst[k], end=' ')
    k += 1
```

a) What will be printed as a result of executing the code segment? Select one correct answer.
❍ A. 0 2 2 3 3 0
❍ B. 0 7 2 5 3 3
❍ C. 0 7 2 5 5 10
❍ D. 1 7 3 5 4 3
❍ E. 7 2 5 3 3 0

b) – d) Fill in the symbols in the flowchart below so that the logic is identical to that of the Python code above.


Stop
Start
lst = [7, 2, 5, 3, 0, 10]
k = 0
n = len(lst) - 1
no
b)
yes
c)
no
yes
d)
print k, lst[k]

Question 5 (15 points)
The function `avg_val` should return the average value in the matrix as a float.

The `data` parameter is a 2D list, or more specifically, a list of lists. `data` will have an equal number of rows and columns, i.e. it has dimension n x n. For example, a 4 x 4 matrix would be defined as follows:

```
data = [[12, 6, 3, -4],
        [5, 2, 7, 16],
        [-9, 10, 11, 1],
        [13, -16, 14, 8]]
```

In this example, the sum of all the cells is 79. You'll want to return 79 / 16, which is 4.9375, the average value of all the cells. Remember, your solution must work for a matrix of arbitrary size n x n.

Here is the matrix:

12	6	3	-4
5	2	7	16
-9	10	11	1
13	-16	14	8

Fill in the blanks to complete this function. Do **not** use the len() or range() functions in any of your answers.

```
def avg_val(data):
    n = len(data)

    total = 0

    for row in ________(a)_________:

        for val in ________(b)_________:

            total += ________(c)_________

    return ________(d)_________ / _______(e)_________
```

Question 6 (10 points)
A class that is derived from another class is called a _________(a)__________, and it _________(b)__________ the attributes and methods from that other class.

In OO programming, _________(c)__________ hides the details of the inner workings of the class from the client. The fields and methods to change their state are contained within the class definition.

The `__init__()` method is also called the _________(d)__________.

In OO programming, a _________(e)__________ instance variable, denoted as `__varname`, cannot be accessed except from inside the class.

Question 7 (10 points)
Consider the following incomplete class definitions:

```
class BankAccount(object):
    def __init__(self, account_num, balance):
        '''Constructs a BankAccount.
           account_num is an integer.
           balance is a float representing US dollars and cents.
        '''
        self.__account_num = account_num
        self.__balance = balance

    @property
    def balance(self):
        return self.__balance

    def withdraw(self, amount):
        self.__balance -= amount
        return self.__balance

    def deposit(self, amount):
        self.__balance += amount
        return self.__balance

    def __str__(self):
        return 'Account: %d, balance $%.2f' % \
               (self.__account_num, self.__balance)

class MinimumBalanceAccount(BankAccount):
    def __init__(self, account_num, balance, minimum_balance):
        '''Constructs a MinimumBalanceAccount.
           account_num is an integer.
           balance is a float representing US dollars and cents.
           minimum_balance is a float representing US dollars and cents.
        '''
        super().__init__(account_num, balance)
        self.__minimum_balance = minimum_balance

    def withdraw(self, amount):
        if self.balance - amount < self.minimum_balance:
            raise ValueError('Minimum balance must be maintained.')
        super().withdraw(amount)

    @property
    def minimum_balance(self):
        __________xxx__________

    __________yyy__________
    def minimum_balance(self, minimum_balance):
        self.__minimum_balance = minimum_balance

    def __str__(self):
        return __________zzz__________ + ', minimum balance $%.2f' % \
               self.__minimum_balance
```

a) Complete the line of code to instantiate a BankAccount with account number 456 and a balance of $130.00.
`bank_acct = ________________________________`

b) The withdraw() method is ________________________ in the MinimumBalanceAccount class.
(a) overloaded (b) abstract (c) overridden (d) overwritten (e) instantiated

c) Write the line of code to replace 'xxx' in property minimum_balance.
____

d) Write the decorator to replace 'yyy' in the minimum_balance setter.
____

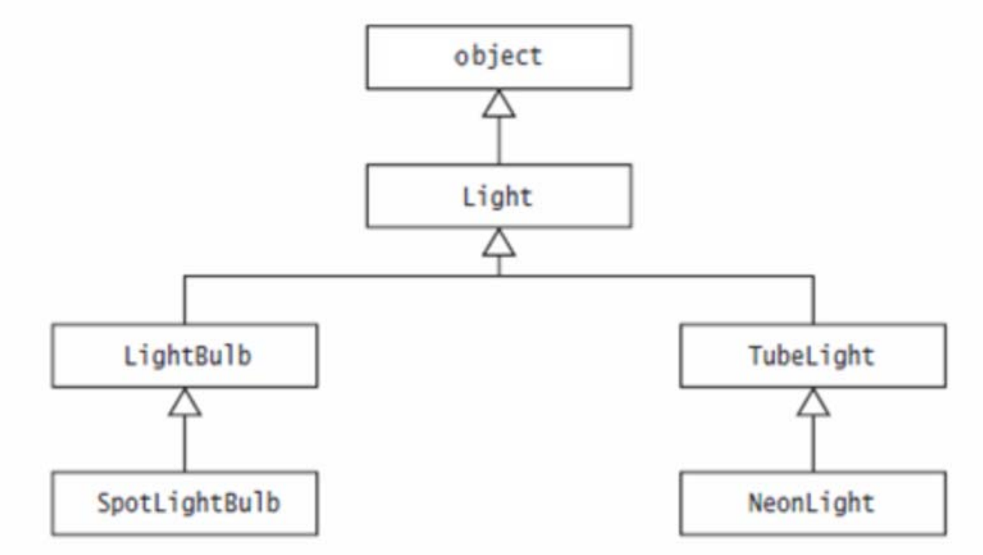
e) Write the missing code to replace 'zzz' in the __str__ method of MinimumBalanceAccount. A MinimumBalanceAccount with account number 6816271837, a balance of $134.56, and a minimum balance of $100.00 should be represented by the string:

```
Account: 6816271837, balance $134.56, minimum balance $100.00
```

To receive credit, you must use the __str__ method in BankAccount.

____

Question 8 (10 points)



Use the inheritance diagram to complete the class definitions below:

```
class SpotLightBulb(_____(a)_____):
    pass

class _____(b)_____(TubeLight):
    pass
```

Assuming we have executed

```
light = NeonLight()
```

what is printed on the screen for the following statements?

```
c) print( isinstance(light, TubeLight) )

d) print( isinstance(light, LightBulb) )

e) print( isinstance(light, Light) )
```