

Question 1

- 1) The first element of a Python tuple is at index position 1 True **False**
- 2) In Python, strings are immutable **True** False
- 3) The split () method provides a way to break a line from file
Into a list of individual words **True** False
- 4) If a == "w" or "W": checks to see if a is equal to "w" or if a is
equal to "W" True **False**
- 5) Tuple are immutable in Python **True** False
- 6) When reading a file in Python, you must specify the file name
and mode **True** False
- 7) When the readline method reaches the end of the file a
"" is returned **True** False
- 8) The keyword def is used to define a method in Python. **True** False
- 9) A statement def __init__(self): would likely be a constructor
in a Python class **True** False
- 10) The self parameter in a method is used to refer to the object
on which the method was invoked **True** False

Question2:

- 1) What is the result of `[5] * 5`?
 - a. **[5 5 5 5 5]**
 - b. `[5 5]`
 - c. 25
 - d. `[25]`
 - e. None of the above are true
- 2) If `names = ["Abby", "Bob", "Carl", "Dan", "Edith"]`. What does `print (names [-3: -1])` print?
 - a. `["Bob", "Carl", "Dan"]`
 - b. **["Carl", "Dan"]**
 - c. `["Abby", "Bob", "Carl"]`
 - d. `[]`
 - e. None of the above
- 3) One key difference between a set and a list is:
 - a. **set elements are not stored in any particular order.**
 - b. List elements are not stored in particular order.
 - c. Set elements can be accessed directly using their position
 - d. Sets are immutable, lists are mutable.
 - e. None of the above
- 4) Which statement correctly creates an empty set flags ?
 - a. `Flags = {}`
 - b. `Flags = set{}`
 - c. **Flags = set()**
 - d. `Flags = null`
 - e. None of the above
- 5) What are the keys in the following dictionary?
`Fruit {"Apple": "Green", "Banana": "Yellow"}`
 - a. **"Apple" and "Banana"**
 - b. "Green" and "Yellow"
 - c. "Apple", "Banana", "Green" and "Yellow"
 - d. The dictionary does not have any keys
 - e. Nothing there is an error

6) Consider the following code segment. What is displayed when it is executed?

```
x = {}  
x["Hello"] = [4,5]  
x["Hello"].append("World")  
print(x["Hello"])
```

- a. "Hello"
 - b. "HelloWorld"
 - c. [4,5]
 - d. [4,5,"World"]**
 - e. None of the above
- 7) Which statement best describes the public interface for a class?
- a. The act of hiding the implementation details.
 - b. A collection of method through which the objects of the class can be manipulated.(经讨论, 参考课本)**
 - c. Enables changes in the implantation without affecting users of a class. (某知名机构老师)
 - d. Instance variables used to store the data required for executing its methods.
 - e. The name of the class.
- 8) Assume class exists named fruit. Which of the follow statements contracts and object of fruit class?
- a. def Fruit():
 - b. class Fruit():
 - c. x = Fruit.create()
 - d. x = Fruit()**
 - e. None of the above.
- 9) Consider the following code segment, What statement should be placed in the blank to complete the constructor for LandVehicle?

Class Vehicle:

```
Def __init__(self,name):  
    Self._name = name
```

Class LandVehicle(Vehicle):

```
Def __init__(self,numWheels):  
    Super().__init__("LandVehicle")
```

```
Def getWheels(self):  
    Return self._numWheels
```

- a. NumWheels = numWheels
- b. self._numWheels = numWheels**
- c. super()._numWheels = numWheels
- d. super().__init__(numWheels)
- e. None of the above.

10) Consider the following code segment and then determine which of the following statement is most correct.

```
Class Fruit:  
    ...  
    def getName(self):  
        ...  
Class Apple(Fruit):  
    ...  
    def getName(self):  
        ...
```

- a. The Apple class overrides the fruit class
- b. The Apple class overrides the getName method.**
- c. The Fruit class overrides the Apple class.
- d. The Fruit class overrides the getName method.
- e. None of the above.

Question 3

- 1) describe what the code below computes and prints?

```
pos1 = -1
pos2 = -1
str = input("Enter a string: ")
ach = input("Enter a character: ")
for i in range(0, len(str)):
    if str[i] == ach:
        if pos1 == -1:
            pos1 = i
            pos2 = i
        else:
            pos2 = i
print(pos1, pos2)
```

Answer:

The code computes the position of the characters which exist in the string and if the character appears in the code twice, it prints the positions of the character in the string. If the character only appears once, the first output will be the position of character in the string and the second output will be the initial value of the variable pos2 which is -1. If the character doesn't appear in the code, the output will be the initial value of the variables which are -1s

code 将会尝试去寻找用户输入的 character 在输入的 string 里存不存在, 如果存在 pos1 和 pos2 将会分别代表第一次和第二次发现 character 的位置。如果输入的 string 内没有输出的 character 存在将会 print 出 -1, -1

- 2) given the following code snippet, what is final version of the list names?

```
names = []
names.append("Amy")
names.append("Bob")
names.append("Peg")
names[1] = "Ruth"
names.pop(2)
names[0] = "Cy"
names.insert(0, "Ravi")
```

Answer: ['Ravi', 'Cy', 'Ruth']

3) what is the output of the following code snippet?

```
fibonacci = {1,1,2,3,5,8}
primes = {2, 3, 5, 7, 11}
both = fibonacci.union(primes)
print(both)
```

Answer: {1, 2, 3, 5, 7, 8, 11}

4) consider the following code segment:

```
fruit = ("Apple": "Green", "Banana": "Yellow")
fruit ["Plum"] = "Purple"
```

after it executes, what is the value of fruit? What are the keys?

Answer: value of fruit: {'Apple': 'Green', 'Banana': 'Yellow', 'Plum': 'Purple'}

Keys: Apple, Banana, Plum

5) the code snippet below, if the file *input.txt* contains the following three lines:

```
Monday!
Tuesday.
Wednesday?
```

What would be the output?

```
infile = open("input.txt", "r")
for word in infile:
    word = word.lstrip("!.")
    print(word)
```

Answer:

Monday!

Tuesday.

Wednesday?

(注意每行字之间有一行空白)

6) consider the following the class hierarchy:

```
class Vehicle:
    def __init__(self, atype):
        self._type = atype
class LandVehicle(Vehicle):
    def __init__(self, atype):
        super().__init__(atype)
class Auto(Vehicle):
    def __init__(self, atype):
```

Complete the code in the *Auto* class constructor to store the *atype* data.

Answer: super().__init__(atype)

7) Consider the class employee:

```
Class Employee:
    Def __init__(self, firstName, lastName, employeeld):
        self._name = lastName + "," + firstName
        self._employeeld = employeeld
```

if an object is constructed as:

```
sam = Employee("Sam", "Fisher", 54321)
```

what is the contents of the instance variable *_name*?

Answer: Fisher,Sam

The following program counts the frequency of words in a file. Answer questions 8-12 about it and revise the code as per the questions.

```
1. filename = input("Enter the name of a file(or quit to quit):")
2. inf = open(filename, "r")
3.
4. counts = {}
5. # Count the words in the file.
6. for line in inf:
7.     words = line.split()
8.     for word in words:
9.         if word in counts:
10.            counts[word] = counts[word] + 1
11.        else:
12.            counts[word] = 1
13.        for word in (counts):
14.            print("%s: %d" % (word, counts[word]))
```

8) What variable is the file object and in what line is it created?

Answer: variable: inf ,

position: Line 2

9) what data structure does this program use?

Answer: dictionary

10) After what line would you include a statement to close the file? Write the statement to do that.

Answer: Line 12 , inf.close()

11) How would you change the code to remove all white space from the start and end of an input line? Write the code to do that and indicate which line it follows or which line you would change.

Answer: Line 7 , words = line.strip().split()

脑残分解法 words = (line.strip(" ")).split() (引号中间空格, 可省略" "整体)

12) What line would you change to print the words in sorted order? Write the new line to do that.

Answer: After Line 12. Code: for word in sorted(counts):

Question 4: logic errors – correcting code segments – 16marks

- 1) The function is to compute the difference between an element in the list and the element following it. It is to return a list of these difference. The program has 3 logic errors. Identify them and correct the lines of code that contain them. *Note:* a line may contain more than one logic error.

```
def difflist(lst):
    newlist = []
    for i in range (1, len(newlist)):    #for i in range(0,len(lst)
        n = lst [i + 1]
        m = lst [i]
        newlist.append(n-m)            # newlist.append(m-n)
    #return newlist (注意一下位置)
```

- 2) The following code is supposed to read lines from a file and determine the longest sequence of the same character in each line. The name of the file and the character to look for in the sequence is input by the user. The program has 5 logic errors: find them and correct them. *Note*: a line may contain more than one logic error.

```
inFile = input("Enter file name:")
inf = open(inFile, "r")
# prompt for character to search for
achar = input("Enter search character: ")

for line in inf:
    line = line.rstrip()
    long = 0
    inRun = False
    nseq = 0
    for i in range(0, len(line)-1):      #(0,len(line))
        if achar == line[i + 1]:      #line[i]
            if inRun:
                nseq = nseq + 1
            else:
                nseq = 1
                inRun = False          #inRun=True
        else:
            if inRun and nseq > long:
                nseq = long          #long = nseq
                inRun = False
                nseq = 0
            else:
                inRun = False
                nseq = 0
    if nseq > long:
        long = nseq
    print("For line: " + line)
    print(" Longest sequence of" + achar + " in the line is ", nseq)
    # print(" Longest sequence of" + achar + " in the line is ", str(long))
inf.close()
```

(这道题不清除倒数第二行算是一个错误还是两个错误如果算两个错误的话正好五个符合要求如果算一个的话那我也找不到剩下那个了。)By Henry

(嘿嘿嘿 5:07am 2017/12/10 找到了最后一个) by P

Question 5:

Menu Class:

Class Menu:

```
def __init__(self) :
    self._options = {}
    self._num_options = 0

def addoptions(self,option):
    self._num_options += 1
    self._options[self._num_options] = option

def displayMenu(self):
    for key in sorted(self._options)
        print("%.3d %10s" %(key, self._options[key]))

def getInput(self):
    done = False
    while not done:
        self.displayMenu()
        userChoice = int(input("Enter Choice:"))
        if userChoice >= 1 and userChoice <= self._num_options :
            done = True
    return userChoice
```

THIS IS PRODUCED BY

Western CS 程序开发有限公司

感谢名单：

群中巨佬 Henry

知名打字 Rocky

真识字员 Jacky

女装大佬 Yuka 酱

Bluekey Neo 老师

某不知名泄露原文件人士

不睡觉的 Peter

以及本群所有英俊帅气的男群员

以及个个美若天仙的女群员

5.1. What is the constructor for the class?

Answer:

```
def __init__(self) :  
    self._options = {}  
    self._num_options = 0
```

5.2. What are instance variables in the class?

Answer:

```
self._options  
self._num_options
```

5.3. Write a 'getter' method that will get the number of options in a menu.

Answer:

```
def getNumOfOptions(self):  
    return self._num_options
```

5.4. Write a method that would get the 'name' of an option, given an option number.

Answer:

```
def getNameOfAnOption(self,opNum):  
    return self._options[opNum]
```

5.5. Write a test program that would a) create a menu object with 3 options, "Option A", "Option B", "Option C", b) prompt the user to enter an option and c) print the user's choice.

Answer:

```
obj = Menu()  
obj.addoptions("Option A"):  
obj.addoptions("Option B"):  
obj.addoptions("Option C"):  
print(obj. getInput())
```

THIS IS PRODUCED BY

Western CS 程序开发有限公司

感谢名单：

群中巨佬 Henry

知名打字 Rocky

真识字员 Jacky

女装大佬 Yuka 酱

Bluekey Neo 老师

某不知名泄露原文件人士

不睡觉的 Peter

以及本群所有英俊帅气的男群员

以及个个美若天仙的女群员

Question 6

The following classes, Contact and Address, have been started to provide that maintains contact lists – names, addresses, phone numbers, email addresses, etc. Review the code provided for these classes and then answer the questions and provide the code required.

Contact Class(partial)

Class Contact:

```
def __init__(self, fName = "", lastName = ""):
    self.firstName = fName
    self.lastName = lastName
    self.email = ""
    self.phone = ""
    self.mobilePhone = ""
    self.address = Address("")

def getName(self):
    return self.firstName + " " + self.lastName

def getFirstName(self):
    return self.firstName

def getLastName(self):
    return self.lastName
```

Address Class(partial)

```
class Address:
    def __init__(self, aline = ""):
        self.addr = aline

    def getFullAddr(self):
        return self.addr

    def addLine(self, aline):
        if self.addr == "":
            self.addr = aline
        else:
            self.addr = self.addr + "\n" + aline
```

6.1. Write a 'setter' method for the Contact class to set a contact's email address.

Answer:

```
def setEmail (self,emailAd):
    self.email = emailAd
```

6.2. Write a 'setter' method for the Contact class to set a contact's address.

Answer:

```
def setAddress(self,address):
    self.address.addLine(address)
```

6.3. Write a method for the Contact class to see if two contacts are equal, that is, they have the same first names, last names and email addresses.

Answer:

```
def __eq__(obj):
    if self.firstName is not obj. firstName:
        return False
    if self. lastName is not obj. lastName:
        return False
    if self. email is not obj. email:
        return False
    return True
```

6.4. Write a test program to test parts of these classes. The program should create a new contact using the information below and then use the methods of the classes to print out the name and address of the contact; it should use your setter method from question 6.2. The information to use is below. Note that the address has (and should have) 3 separate lines

First name = "Mickey"

Last name = "Mouse"

Address = "Disney World
1 Magic Kingdom Road
Orlando, Florida"

Answer:

```
obj = Contact("Mickey", "Mouse")
obj.setAddress("Disney World")
obj.setAddress("1 Magic Kingdom Road")
obj.setAddress("Orlando, Florida")
print(obj.getName())
print(obj.address.getFullAddr())
```

THIS IS PRODUCED BY

Western CS 程序开发有限公司

感谢名单：

群中巨佬 Henry

知名打字 Rocky

真识字员 Jacky

女装大佬 Yuka 酱

Bluekey Neo 老师

某不知名泄露原文件人士

不睡觉的 Peter

以及本群所有英俊帅气的男群员

以及个个美若天仙的女群员