

CS1026 Computer Science Fundamentals Quiz 2

1. The function below is to compute the difference between an element in the list and the element following it, i.e., subtracts the second element from the first. It is to return a list of these differences. The program has 7 logic errors. Identify them and correct the lines of code that contain them. *Note:* a line may contain more than one logic error.

<pre>def difflist(lst): newlist = [] for i in range(1, len(newlist)): n = newlist[i+1] m = newlist[i] newlist.append(n-m)</pre>	<pre>def difflist(lst): newlist = [] for i in range(0, len(lst)-1): #0 #lst #lst - 1 n = lst[i+1] #lst m = lst[i] #lst newlist.append(m-n) #m-n return newlist #return newlist</pre>
---	--

2. In the code snippet below, if the file `input.txt` contains the following three lines:

Monday!
Tuesday.
Wednesday?
What would be the output?

<pre>infile = open("input.txt", "r") for word in infile: word = word.lstrip("!.") print(word)</pre>	Output: Monday! Tuesday. Wednesday?
---	--

3. Consider the following code segment:

```
print("W", end="")
try:
    inFile = open("test.txt", "r")
    line = inFile.readline()
    value = int(line)
    print("X", end="")
except IOError:
    print("Y", end="")
except ValueError:
    print("Z", end="")
```

What output is generated when this program runs if `test.txt` is not opened successfully? WY

4. 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)
{1, 2, 3, 5, 7, 8, 11}
```

5. 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?
{'Apple': 'Green', 'Banana': 'Yellow', 'Plum': 'Purple'}
Keys: Apple, Banana, Plum
```

6. The following class, `Message`, is a class that might be used in designing an email system. Review the class definition and then answer the questions that follow.

```
class Message :
    def __init__(self, sender, recipient) :
        self._sender = sender
        self._recipient = recipient
        self._body = ""
    def append(self, line) :
        self._body = self._body + line + "\n"
    def __str__(self) :
        result = "From: " + self._sender + "\n"
        result = result + "To: " + self._recipient + "\n"
        result = result + self._body
        return result
```

- a. What is the constructor for the class `Message`?

```
__init__(self, sender, recipient)
```

- b. What are the instance variables for this class?

```
_sender, _recipient, _body
```

- c. Explain what the method `__str__` does and how it is used.

Defines how to print an object of this Class and is used in the Python “print” command if an object of class Message is included for printing. The “print” command then uses `__str__` to print the object.

- d. Write a “getter” method for the sender instance variable

```
def getSender(self) :
    return self._sender
```

- e. Write a “setter” method for the sender instance variable

```
def setSender(self,newSender) :
    self._sender = newSender
```

7. The following class, `Mailbox`, makes use of the class `Message` in the previous question and stores a collection of messages. Review the code that is given and then answer the questions that follow.

```
from message import Message
class Mailbox :
    def __init__(self) :
        self._messages = []
    def addMessage(self, message) :
        self._messages.append(message)
    def getMessage(self, index) :
        return self._messages[index]
    def print(self) :
        for i in range(len(self._messages)):
            print(self._messages[i])
            print()
```

- a. What is the constructor for the class `Mailbox`?

```
__init__(self)
```

- b. What are the instance variables in the class `Mailbox`?

```
_messages
```

- c. What data structure is used to store messages?

List

d. Write a method `removeMessage` that removes a message from a mailbox given its index number.

```
def removeMessage(self, index) :  
    self._messages.pop(index)
```

e. Write a main program that will a) create a new mailbox, b) prompt a user for a sender, a receiver and multiple lines of a message, and c) will create a message of class `Message` and add it to the mailbox.

```
from mailbox import Mailbox  
from message import Message  
  
# Create a mailbox and messages to it.  
mbox = Mailbox()  
sto = input("Enter who the message goes to: ")  
sfrom = input("Enter who the message is from: ")  
msg = Message(sfrom, sto)  
aline = input("Enter a line of the message: ")  
while aline != "":  
    msg.append(aline)  
    aline = input("Enter a line of the message: ")  
mbox.addMessage(msg)
```