

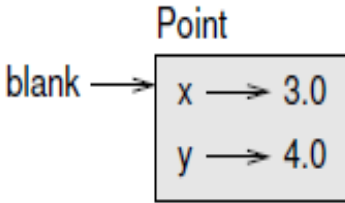
Internal Assessment Test 3 – Dec 2022

Sub:	Python Application Programming				Sub Code:	18CS742	Branch:	ECE/EEE/MECH	
Date:	28/12/2022	Duration:	90 mins	Max Marks:	50	Date:	28/12/2022		Duration:

Answer any FIVE FULL Questions

MARKS

CO RBT

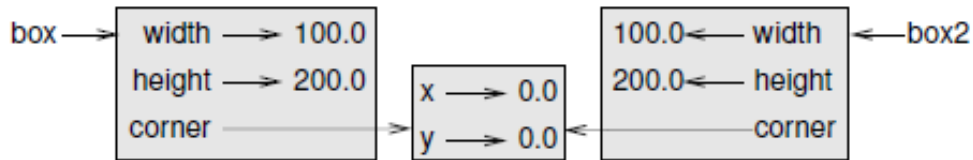
1(a)	<p>Write a class for the point object. Use <code>__init__</code> and <code>__str__</code> to instantiate and print an instance of point. Expected output: (3.0,4.0)</p> <div style="text-align: center;"></div> <pre>class Point: def __init__(self, x,y): self.x=float(x); self.y = float(y); def __str__(self): return f'({self.x},{self.y})' p=Point(5,6) print(p)</pre>	[04]	CO4	L3
1(b)	<p>What is a pure function? Write a pure function to calculate distance between two points using the point object created in part (a) $d(P,Q)=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$</p> <pre>def distance(p1,p2): #write code here</pre> <p>Answer:</p> <ul style="list-style-type: none">• Does not modify any of the objects passed to it as arguments.• It has no effect<ul style="list-style-type: none">○ Getting user input○ Displaying a value• It only returns. <pre>import math def distance(p1, p2): return math.sqrt((p1.x-p2.x)**2 + (p1.y-p2.y)**2) p1=Point(3,2) p2 =Point(10,5) print(f'Distance between {p1},{p2}={distance(p1,p2):0.2f}')</pre>	[06]	CO4	L3

2 (a) Differentiate deep and shallow copy with code snippets.

[07]

CO4

L2



- Shallow copy.
 - it copies the rectangle object.
 - But **does not** copy the **embedded object**.
 - Called a **shallow copy: copies object and its references**
- >>> box.corner.x,box.corner.y=0.0,0.0
- >>> box.width,box.height=100,200
- >>> box_copy = copy.copy(box)
- >>> box is box_copy
- False
- >>> box.corner is box_copy.corner
- True

Deep Copy:

copies not only the **object but also the objects it refers to,**

```
>>> box_dc = copy.deepcopy(box)
```

```
>>> box is box_dc
```

False

```
>>> box.corner is box_dc.corner
```

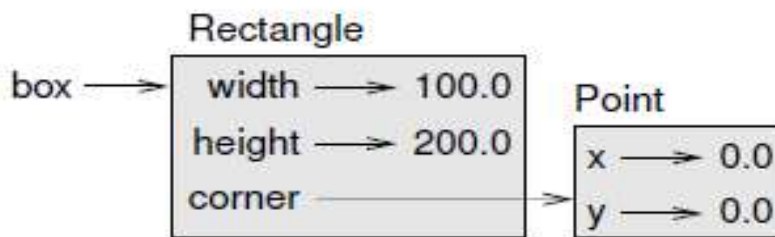
False

(b)

[03]

CO4

L3



What is the output and why?

```
>>> import copy
```

```
>>> box_c = copy.copy(box)
```

```
>>> box is box_c
```

```
>>> box.corner is box_c.corner
```

False

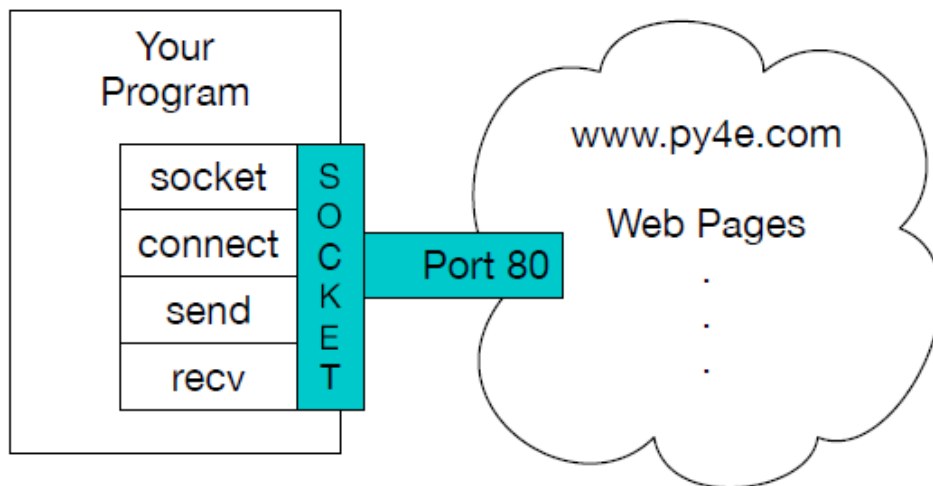
True

- Shallow copy **does not** copy the **embedded object**.

3	Define polymorphism.	3M	CO4	L2
b	<p>Demonstrate polymorphism with function to find histogram to count the number of times each letter appears in a word and use same function for lists or tuples.</p> <ul style="list-style-type: none"> • Type-based dispatch- not always necessary • Functions can be written that work correctly for arguments with different types • Functions that can work with several types are called polymorphic • facilitate code reuse • Sum built in function can be used to add because Time can add two objects. • <code>print(sum([t1,t2,t3]))</code> <p>Expected Output:</p> <pre>>>> h = histogram('brontosaurus') >>> h {'a': 1, 'b': 1, 'o': 2, 'n': 1, 's': 2, 'r': 2, 'u': 2, 't': 1} >>> t = ['spam', 'egg', 'spam', 'spam', 'bacon', 'spam'] >>> histogram(t) {'bacon': 1, 'egg': 1, 'spam': 4} def histogram(a): freq={} for item in a: freq[item] = freq.get(item,0) +1 return freq h = histogram('brontosaurus') print(h) t = ['spam', 'egg', 'spam', 'spam', 'bacon', 'spam'] h = histogram(t) print(h)</pre>	7M	CO4	L3
4	<p>Explain operator overloading in Python.</p> <ul style="list-style-type: none"> • Changing the behavior of an operator so that it works with user-defined types is called • There are special methods that specify behaviour of operators on user defined types • <code>__add__</code> will be called if you use the on two objects. • def __add__(self, other): • sum = self.time_to_seconds() + other. + operator time_to_seconds() 	[5]	CO4	L2

	<ul style="list-style-type: none"> • return self.seconds_to_time(sum) 			
	<p>Demonstrate operator overloading the '+' and '*' operator to perform addition and multiplication of two complex numbers</p> <p>OR</p> <p>'+' operator on the TIME object for calculating t1 + t2</p> <p>class Time:</p> <pre> def __init__(self, hour=0, minute=0, second=0): self.hour=hour self.minute = minute self.second = second def __str__(self): return f'{self.hour:02d}:{self.minute:02d}:{self.second:02d}' class Time: def __init__(self, hour=0, minute=0, second=0): self.hour=hour self.minute = minute self.second = second def __str__(self): return f'{self.hour:02d}:{self.minute:02d}:{self.second:02d}' def convertToSec(self,t): return t.second+t.minute*60+t.hour*60*60 def convertToTime(self,sec): m,s=divmod(sec,60) h,m=divmod(m,60) return Time(h,m,s) def __add__(self,other): total = self.convertToSec(self) +self.convertToSec(other) return self.convertToTime(total) t1 = Time(3,45,20); t2 = Time(2,40,50) print(t1+t2) </pre>	5M		
5	<p>Define a socket with a neat diagram</p> <p>Built in support in Python : socket</p> <p>Socket : similar to a file</p> <p>But it provides two way connection between two programs</p> <p>You can read from and write to the same socket.</p>	[04]	CO5	L2

Write to a socket : it is sent to the application on the other side
 Read from a socket : get the data sent by the application
 Some coordination is required – hence a protocol



b

Demonstrate with the help of suitable code snippet

i) To retrieve a plain text file over HTTP.
 ii) To retrieve web pages with urllib.

```

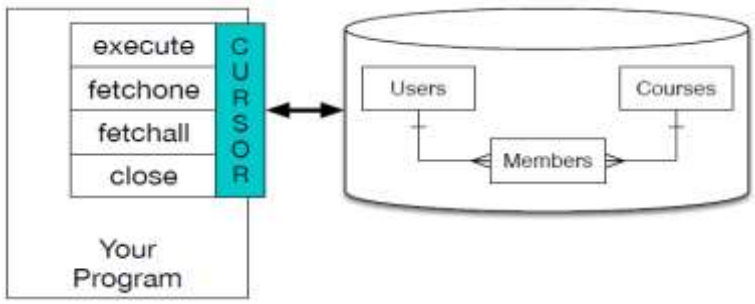
import socket
mysock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
mysock.connect(('data.pr4e.org',80))
cmd = 'GET http://data.pr4e.org/romeo.txt HTTP/1.0\r\n\r\n'.encode()
mysock.send(cmd)
while True:
    data = mysock.recv(512)
    if len(data)<1:
        break
    print(data.decode(), end="")

mysock.close()

import urllib.request as ur
fhand = ur.urlopen('http://data.pr4e.org/romeo.txt')
for line in fhand:
    print(line.decode().strip())
  
```

Output:
 python browser_ul.py
 But soft what light through yonder window breaks It is the east and Juliet is the sun
 Arise fair sun and kill the envious moon Who is already sick and pale with grief

6M CO5 L3

6(a)	<p>Explain cursor with a neat diagram.</p> <ul style="list-style-type: none"> • a file handle that we can use to perform operations on the data stored in the database • cursor() in database is like open() when dealing with text files  <p>The diagram illustrates a cursor object on the left, labeled 'Your Program'. It contains four methods: 'execute', 'fetchone', 'fetchall', and 'close'. A vertical bar next to these methods is labeled 'CURSOR'. A double-headed arrow connects this cursor object to a database cylinder on the right. Inside the database cylinder, there are three tables: 'Users', 'Courses', and 'Members'. 'Users' and 'Courses' are connected to 'Members' by lines, indicating relationships.</p>	[03]	CO5	L2												
6(b)	<p>Write a python program to create a database moviesDB in sqlite.</p> <ol style="list-style-type: none"> Drop existing table if any Create a table with the following schema: movie(title(text), length(number), rating(number)) Insert the following items <table border="1" data-bbox="183 851 973 1019"> <thead> <tr> <th>Title</th> <th>Length (minutes)</th> <th>Rating</th> </tr> </thead> <tbody> <tr> <td>The Matrix</td> <td>136</td> <td>8.7</td> </tr> <tr> <td>Lord of the Rings</td> <td>201</td> <td>8.8</td> </tr> <tr> <td>The Martian</td> <td>142</td> <td>8</td> </tr> </tbody> </table> <ol style="list-style-type: none"> Query the table and sort by highest to lowest rating. Display the details of each row. <pre> import sqlite3 conn = sqlite3.connect('movies.sqlite') cur = conn.cursor() cur.execute('DROP TABLE IF EXISTS Movie') cur.execute('CREATE TABLE Movie (title VARCHAR(100),length INTEGER,rating DECIMAL)') cur.execute("INSERT INTO Movie VALUES ('The Matrix',136, 8.7)"); cur.execute("INSERT INTO Movie VALUES ('Lord of the Rings',201, 8.8)"); cur.execute("INSERT INTO Movie VALUES ('The Martian',142, 8)"); cur.execute('SELECT * FROM MOVIE ORDER BY rating DESC') for row in cur: print(row) conn.close() python iat3_db.py ('Lord of the Rings', 201, 8.8) ('The Matrix', 136, 8.7) </pre>	Title	Length (minutes)	Rating	The Matrix	136	8.7	Lord of the Rings	201	8.8	The Martian	142	8	[07]	CO5	L3
Title	Length (minutes)	Rating														
The Matrix	136	8.7														
Lord of the Rings	201	8.8														
The Martian	142	8														

	('The Martian', 142, 8)			
--	-------------------------	--	--	--

CI

CCI

HOD

Course Outcomes		PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3	PSO 4
		CO1	1	3	2	2		3				2	2				
CO2	1, 2	3	2	2		3				2	2						
CO3	3	3	3	3	2	3				2	2					2	2
CO4	4	3	2	3		3				2	2						
CO5	5	3	2	3	3	3				2	2			3	2	2	2
