DRAFT Scheme of Valuation/Answer Key

(Scheme of evaluation (marks in brackets) and answers of problems/key)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

SIXTH SEMESTER B.TECH DEGREE (S) EXAMINATION, May 2023

Course Code: CST362

Course Name: PROGRAMMING IN PYTHON

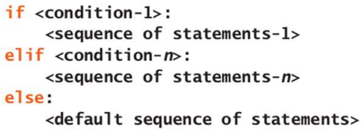
Max. Marks: 100 Duration: 3 Hours

PART A

Answer all questions, each carries 3 marks. Marks

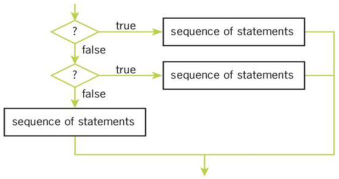
1 Output: 0,1,2,3 + Justification (1.5 +1.5) (3)

2 Multiway-if Syntax - 1.5 (3)



```
if <condition-1>:
    <sequence of statements-1>
elif <condition-n>:
    <sequence of statements-n>
else:
    <default sequence of statements>
```

Semantics - 1.5



3 Definitions of namespace, scope and lifetime - 1*3 =3 (3)

4 Mutable and immutable properties of Python data structures -1 (3)

Any mutable Data structure (eg. List, Dictionary etc) – 1

Any immutable data structure (eg. string, integer, tuple etc) - 1

5 Attributes of a turtle object- 3 (3)

1. Location

2. Orientation (or direction), and

3. Pen (with attributes color, width, and on/off state)

6 Advantages of GUI based programs over terminal based programs. 3*1 =3 (3)

1.The user is not constrained to enter inputs in a particular order.

2. Running different data sets does not require re-entering all of the data.

3. As the number of command options increases and the

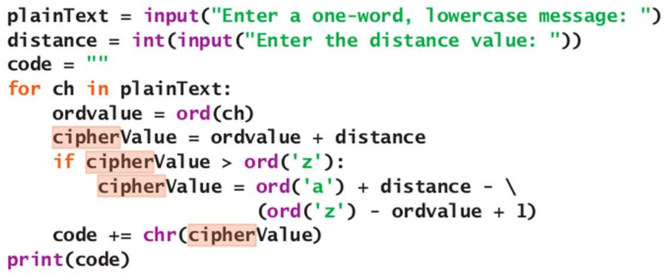
		information to be presented grows in quantity and complexity GUI becomes more useful.	
7		Abstraction mechanism Explanation -2 - They simplify design and controlling the complexity of solutions. - It gives user an exteranl view of a resource, showing what it does and how it can be used -Programmer shouldn’t be concerned with how a resource performs its task. Example -1	(3)
8		Definitions of accessors & mutators - 1.5 each - Methods that allow a user to observe but not change the state of an object are called accessors - Methods that allow a user to modify an object’s state are called mutators .	(3)

9		import os
		file_exists = exists(path_to_file)
		OR
		import os
		file_exists=os.path.isfile('./final_data.csv')
			(3)

10		Flask is a web application framework written in Python. + Explanation -1.5 Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine. + Explanation -1.5	(3)
PART B			
Answer one full question from each module, each carries 14 marks.			
		Module I	
11	a)	Input reading - 1 Real roots implementation – 3 Imaginary root implementation -2 Display of output - 1	(7)
	b)	Input reading - 1	(7)

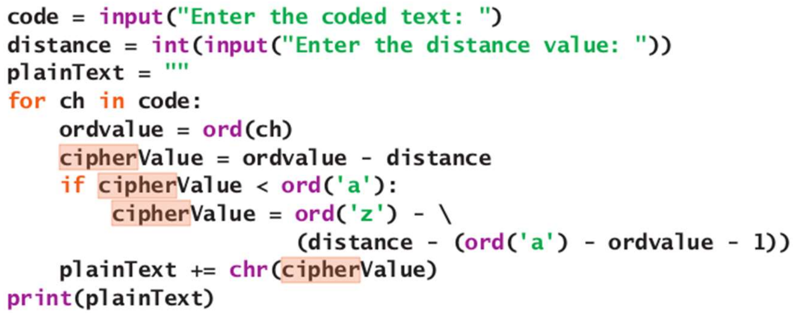
<table>
<tr><td></td><td></td><td>Armstong number logic implementation -5
Display of output - 1</td><td></td></tr>
<tr><td></td><td></td><td><b>OR</b></td><td></td></tr>
<tr><td>12</td><td>a)</td><td>Input reading - 1
Logic implementation for Sum of odd numbers between a programmer specified upper and lower limit – 5
Display of output - 1</td><td>(7)</td></tr>
<tr><td></td><td>b)</td><td>Input reading - 1
Series sum logic implementation – 5
Display of output - 1</td><td>(7)</td></tr>
<tr><td></td><td></td><td><b>Module II</b></td><td></td></tr>
<tr><td>13</td><td>a)</td><td>

Encryption: 3.5



```
plainText = input("Enter a one-word, lowercase message: ")
distance = int(input("Enter the distance value: "))
code = ""
for ch in plainText:
    ordvalue = ord(ch)
    cipherValue = ordvalue + distance
    if cipherValue > ord('z'):
        cipherValue = ord('a') + distance - \
                      (ord('z') - ordvalue + 1)
    code += chr(cipherValue)
print(code)
```

Decryption: 3.5



```
code = input("Enter the coded text: ")
distance = int(input("Enter the distance value: "))
plainText = ""
for ch in code:
    ordvalue = ord(ch)
    cipherValue = ordvalue - distance
    if cipherValue < ord('a'):
        cipherValue = ord('z') - \
                      (distance - (ord('a') - ordvalue - 1))
    plainText += chr(cipherValue)
print(plainText)
```

</td><td>(7)</td></tr>
<tr><td></td><td>b)</td><td>

```
count = 0
inputFile = open("myfile.txt", 'r')
for line in inputFile:
    wordlist = line.split()                    File opening+line fetching -4
for word in wordlist:
    if len(word) == 4:
        count += 1
print("There are", count, "lines.")                 Counting     -3
```

</td><td>(7)</td></tr>
</table>

OR

14 a) (7)

```
1. def mean(lyst):
    theSum = 0
   if len(lyst) == 0
     return 0
    for number in lyst:
        theSum += number
    return theSum / len(lyst)                                         -2
2. mode
 def mode(lyst)
    if len(lyst) == 0
        return 0
    lyst.sort()
    midpoint = len(lyst) // 2
    print("The median is", end = " ")
     if len(lyst) % 2 == 1:
          return(lyst[midpoint])
     else:
          return((lyst[midpoint] + lyst[midpoint - 1]) / 2)            -3


3. Mode
def mode(lyst):
  counts = {}
  for item in lyst:
      if item in counts:
          counts[item] += 1
      else:
          counts[item] = 1

  return [key for key in counts.keys() if counts[key] ==  max(counts.values())]
                                                                    -2
```

OR

```
lyst = [10, 30, 50, 10, 50, 80, 50]
print("Mode of lyst is % s" % (max(set(lyst), key = lyst.count))        -2
```

b) **# Using loop + list slicing** (7)

```
test_list = [5, 6, 3, 8, 2, 1, 7, 1]
sublist = [8, 2, 1]

res = False
for idx in range(len(test_list) - len(sublist) + 1):
     if test_list[idx : idx + len(sublist)] == sublist:
        res = True
        break
print("Is sublist present in list ? : " + str(res))                    -7
```

OR

# using all()

```
test_list = [9, 4, 5, 8, 10]
sub_list = [10, 5, 4]

flag = 0
if(all(x in test_list for x in sub_list)):
    flag = 1

if (flag):
    print("Yes, list is subset of other.")
else:
    print("No, list is not subset of other.")                    -7
```

Module III

15 a) (7)

```
import turtle
t = turtle.Turtle()
t.fillcolor("red")
t.pencolor("black")
t.begin_fill()
for count in range(6):
  t.forward(length)
  t.left(60)
t.end_fill()
turtle.exitonclick()                                        - 4
```

<table>
<tr><td></td><td></td><td>turtle object methods used in the code -3</td><td></td></tr>
<tr><td></td><td>b)</td><td><pre>
def blackAndWhite(image):
  blackPixel = (0, 0, 0)
  whitePixel = (255, 255, 255)
  for y in range(image.getHeight()):
        for x in range(image.getWidth()):
            (r, g, b) = image.getPixel(x, y)
             average = (r + g + b) // 3
             if average < 128:
                image.setPixel(x, y, blackPixel)
             else:
                image.setPixel(x, y, whitePixel)

from images import Image
def main(filename = "smokey.gif"):
  image = Image(filename)
  print("Close the image window to continue.")
  image.draw()
  blackAndWhite(image)
  print("Close the image window to quit.")
  image.draw()

if __name__ == "__main__":
main()
</pre>- 4
Explanation of methods - 3</td><td>(7)</td></tr>
<tr><td></td><td></td><td><b>OR</b></td><td></td></tr>
<tr><td>16</td><td>a)</td><td>1) __init__ method with - 6
Easy frame __init__ call
Two addLabel and addFloatField method calls
Two addButton method calls</td><td>(10)</td></tr>
</table>

		2) 2 event handling methods - 4 To compute amount in Euro from amount in Rupees To compute amount in Rupees from amount in Euros	
	b)	Attributes: -2 title (an empty string by default) width and height in pixels resizability (true by default) background color (white by default) Changing the attributes: -2 Any two out of the following three ways 1.through _init__ method 2. reset them in the window's attribute dictionary 3. run a method included in the EasyFrame class	(4)
		Module IV	
17	a)	Rectangle Class defenition with constructor to set height, width - 3 Two member functions to find area, and perimeter -2 + 2	(7)
	b)	Inheritance implementation -Explanantion+ Illustration -4 Each class below the topmost one inherits attributes and behaviors from its ancestors and extends these with additional attributes and behavior. class <new class name>(<existing parent class name>): Polymorphism implementation - Explanantion+ Illustration-3 Two methods that have the same header but have different definitions in different classes.	(7)
		OR	
18	a)	Student class definition with constructor for receiving name and roll number. 2 Methods to : 1. Display - It should display all informations of the student. -1 2. setAge - It should assign age to student -2 3. setTestMarks - It should assign marks of a test to the student. -2	(7)
	b)	Exceptions Explanation and common exceptions in Python-2 Python Try Catch statement -3	(7)

		Illustartion - 2	
		Module V	
19	a)	1. arr2d[:2] = > [[1, 2, 3], [4, 5, 6]] 2. arr2d[:2, 1:] = > [[2, 3], [5, 6]] 3. arr2d[1, :2] => [4, 5] 4. arr2d[:2, 1:] = 0 => [[1, 0, 0], [4, 0, 0], [7, 8, 9]] 2*4=8	(8)

	b)	import csv
		
		fields = ['Reg. No', 'Name', 'Semester', ‘College’,'CGPA'] - reading 3
		
		rows = [[‘ABC123’,’Ganesh Kumar’,’S8’, ‘ABC’,‘9.8’]
		[‘ECH265’,’John Mathew’,’S7’, ‘ECH’,‘9.9’],
		....
		all data rows to be written here in this format]
		
		filename = "university_topper.csv"
		
		with open(filename, 'w') as csvfile: - writing 3
		
		csvwriter = csv.writer(csvfile)
		csvwriter.writerow(fields)
		csvwriter.writerows(rows)
			(6)

		OR	
20	a)	Import panda and read csv - 2 Number of rows and columns => use shape() -1 First five rows => use head() -1	(4)
	b)	Import required libraries, matplotlib library for visualization and importing csv library for reading CSV data. 1. Open the file using open( ) function with ‘r’ mode (read-only) from CSV library and read the file using csv.reader( ) function. 2. Read each line in the file using for loop. 3. Append required columns of the CSV file into lists.	(10)

		4. After reading the whole CSV data, plot the required data as scatter/plot using plt functions. 5*2 =10	
****			