Marking Scheme for COMP Prelim Practical Exam P2 2022

Name : _____ ( ) Class: _____

Task1 (10)	Task2 (10)	Task3 (10)	Task4 (20)	Total (50)

Task 1			
No	Description	M	A
1.	Any **one** from: • = COUNTA(A4:A18) • = COUNTA(B4:B18) • = COUNT(C4:C18) • = COUNT(D4:D18)	[1]	
2.	Any **one** from: • = MODE.SNGL(D4:D18) • = MODE(D4:D18)	[1]	
3.	**One** mark for finding average, **one** mark for rounding up Any **one** from: • =ROUNDUP(AVERAGE(C4:C18),0) • =CEILING.MATH(AVERAGE(C4:C18),1) • =CEILING(AVERAGE(C4:C18),1)	[2]	
4.	**Three** marks for working top formula, **one** mark for rest On top formula, **one** mark for showing Captain for tallest student, **one** mark for showing Vice Captain for second tallest student, and **one** mark for showing Member for all other students. =IF(D4=MAX($D$4:$D$18),"Captain",IF(D4=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D5=MAX($D$4:$D$18),"Captain",IF(D5=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D6=MAX($D$4:$D$18),"Captain",IF(D6=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D7=MAX($D$4:$D$18),"Captain",IF(D7=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D8=MAX($D$4:$D$18),"Captain",IF(D8=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D9=MAX($D$4:$D$18),"Captain",IF(D9=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D10=MAX($D$4:$D$18),"Captain",IF(D10=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D11=MAX($D$4:$D$18),"Captain",IF(D11=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D12=MAX($D$4:$D$18),"Captain",IF(D12=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D13=MAX($D$4:$D$18),"Captain",IF(D13=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D14=MAX($D$4:$D$18),"Captain",IF(D14=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D15=MAX($D$4:$D$18),"Captain",IF(D15=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D16=MAX($D$4:$D$18),"Captain",IF(D16=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D17=MAX($D$4:$D$18),"Captain",IF(D17=LARGE($D$4:$D$18,2),"Vice Captain","Member")) =IF(D18=MAX($D$4:$D$18),"Captain",IF(D18=LARGE($D$4:$D$18,2),"Vice Captain","Member"))	[4]	
5.	**Two** marks if the text in the row with shortest member changes to red. No marks if conditional formatting is not used.	[2]	

Task 2			
6.	`# replace num_rider = 10` **`num_rider = int(input("Enter number riders: ")`**	[1]	

7.	num_rider = int(input("Enter number riders: ")
	min_height = 1.4
	for p in range(num_rider):
	  name = input("Enter name of rider: ")
	  height = float(input("Enter height of rider: "))
	  if height >= min_height:
	  print("Rider is tall enough to ride a pony.")
	  # Output message if rider is less than 1m
	  elif height < 1:
	  print("Rider is not tall enough to ride a pony.")
		[1]	

	# Get input if accompanied by adult if height more than 1m **else:** **adult = input("Is the rider accompanied by an adult? (Y/N) ")**	[1]	
	# Output correct message if accompanied by adult **if adult == "Y":** **print("The rider is not tall enough but is accompanied by an adult to ride the pony.")**	[1] [1]	
	# Output correct message if not accompanied by adult **else:** **print("The rider is not tall enough and is not accompanied by an adult to ride a pony.")**	[1]	

8.	# initialize variable
	riders = []
	num_rider = 10
	min_height = 1.4
	for p in range(num_rider):
	  name = input("Enter name of rider: ")
	  height = float(input("Enter height of rider: "))
	  if height >= min_height:
	  print("Rider is tall enough to ride a pony.")
	  # add rider's name to list
	  riders = riders + [name,]
	  elif height < 1:
	  print("Rider is not tall enough to ride a pony.")
	  else:
	  adult = input("Is the rider accompanied by an adult? (Y/N) ")
	  if adult == "Y":
		[1]
		[1]	

```
            print("The rider is not tall enough but is accompanied
              by an adult to ride the pony.")
            # add rider's name to list
            riders = riders + [name,]
         else:
            print("The rider is not tall enough and is not
              accompanied by an adult to ride a pony.")
# outputs the list of riders
print(riders)
```

[1]

[1]

Task 3

9.

```
word = input("Please enter your word: ")
word = word.lower()
begin_cap = word.isupper()
has_symbols = not word.isalnum()
has_digits = True
for c in words:
   if c.digit():
      has_digits = True
      break
word_len = word.length()
if not begin_cap and not has_symbols and not has_digits:
   if word_len < 3:
      print("You entered a short word.")
   elif word_len <= 8:
      print("You entered a medium word.")
   elif:
      print("You entered a long word.")

if begin_cap:
   print("Error. You entered a word that does not start with a
    capital letter.")
elif has_symbols:
   print("Error.  You  entered  a  word  that  contain  special
    characters.")
elif has_digits:
   print("Error. You entered a word that contain digits.")
```

Corrected lines

```
# shift line 2 with line 3
begin_cap = word.isupper()
word = word.lower()
```

[1]

```
begin_cap = word[0].isupper()
```

[1]

```
has_digits = False
```

[1]

```
for c in word:
```

[1]

	`    if c.isdigit():`	[1]	
	`word_len = len(word)`	[1]	
	`# remove not in front of begin_cap` `if begin_cap and not has_symbols and not has_digits:`	[1]	
	`    if word_len <= 3:`	[1]	
	`    ~~elif~~ else:`	[1]	
	`if not begin_cap:`	[1]	

Task 4 – Develop/Write a program			
10.	User defined function	[1]	
	…… with 2 arguments in the correct order given in question	[1]	
	Management of for loop	[1]	
	…… that repeats for every character in the message	[1]	
	Management of for loop	[1]	
	…… that repeats `shift_down` by `pos` number of times.	[1]	
	Returns the correct output	[1]	

	Possible solution:
	`def encrypt(message,pos):`
	`    output = ""`
	`    for c in message:`
	`        for i in range(pos):`
	`            c = shift_down(c)`
	`        output += c`
	`    return output`		

11.	User-defined function with one argument	[1]	
	Shifts a letter up	[1]	
	Ignores characters other than letters	[1]	

	Possible solution:
	`def shift_up(c):`
	`    if c == "a":`
	`        return "z"`
	`    elif c == "A":`
	`        return "Z"`
	`    elif c.isalpha():`
	`        return chr(ord(c)-1)`
	`    else:`
	`        return c`		

12.	User defined function with 2 arguments in the correct order given in question	[1]	
	Management of for loops	[1]	
	…… that repeats `shift_up` by `pos` number of times for every character in `message`.	[1]	
	Returns the correct output	[1]	
	Possible solution:		

	`def decrypt(message,pos):`
	`  output = ""`
	`  for c in message:`
	`  for i in range(pos):`
	`  c = shift_up(c)`
	`  output += c`
	`  return output`		

13.	Input 'E' to encrypt and 'D' to decrypt	[1]	
	……repeats receiving inputs until a valid input is received	[1]	
	Requests user to enter a message	[1]	
	Requests user to input number of positions to shift	[1]	
	……repeats receiving inputs until a positive whole number is received	[1]	
	Outputs the encrypted/decrypted message	[1]	

	Possible solution:
	`action = input("Please enter 'E' to encrypt or 'D' to decrypt: ")`
	`while action not in ['E','e','D','d']:`
	`  action = input("Error! Please enter 'E' to encrypt or 'D' to decrypt: ")`
	
	`if action in ['E','e']:`
	`  message = input("Please enter message: ")`
	`else:`
	`  message = input("Please enter ciphertext: ")`
	
	`pos = input("Please enter number of positions: ")`
	`while not pos.isdigit() or int(pos)<0:`
	`  pos = input("Error! Please enter number of positions: ")`
	`pos = int(pos)`
	`if action in ['E','e']:`
	`  print("The encrypted text is '{}'" .format(encrypt(message,pos)))`
	`else:`
	`  print("The decrypted text is '{}'" .format(decrypt(message,pos)))`		

	Total	**50**	