		Thumb drive No: STL
Class/ Index Number /	Centre Number/ 'O' Level Index Number /	Name

新加坡海星中学

MARIS STELLA HIGH SCHOOL
PRELIMINARY EXAMINATION
SECONDARY FOUR

COMPUTING **7155/02**
Paper 2 Practical (Lab-based) **26 August 2022**
2 hours 30 mins

Additional Materials:
Electronic version of LIBRARY.XLSX file
Electronic version of NUMBER.PY file
Electronic version of WORD.PY file
Electronic version of CONVERTER.PY file
Insert Quick Reference for Python

READ THESE INSTRUCTIONS FIRST

MARK SCHEME

For Examiner's Use
50

This document consists of **10** printed pages.

Task 1

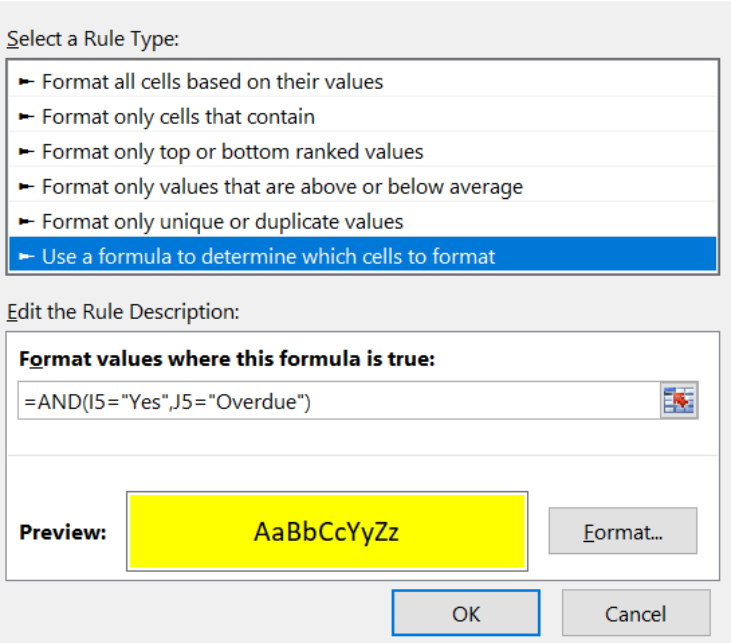
Question	Answer	Marks
1	=LEFT(VLOOKUP(D5, $A$28:$B$40, 2, FALSE),6) & A5 2 marks: VLOOKUP 1 mark: LEFT	3

Question	Answer	Marks
2	=G5-F5	1

Question	Answer	Marks
3	=IF(H5>21, "Overdue", "Not Overdue") Minus 1 mark if message is of wrong spelling/case	2

Question	Answer	Marks
4	=COUNTA(A5:A21)	1

Question	Answer	Marks
5	=COUNTIF(J5:J21, "Overdue")	1

Question	Answer	Marks
6	=AND(I5="Yes",J5="Overdue")  	2

Task 2

Question	Answer	Marks
7	total = int(input('Enter total numbers to input: ')) #6a for count in range(total):	2
8	while number < 1 or number > 100: #6b range check number = int(input('Out of range, please re-enter a positive integer: '))	2

9	total = int(input('Enter total numbers to input: '))
	highlist = [] #6c
	
	for count in range(total):
	number = int(input('Enter a positive integer: '))
	
	while number < 1 or number > 100: #6b range check
	number = int(input('Out of range, please re-enter a positive integer: '))
	
	if number >= 1 and number <= 40:
	print(number, '- Low')
	elif number > 40 and number <= 70:
	print(number, '- Medium')
	elif number > 70 and number <= 100:
	print(number, '- High')
	highlist.append(number) #6c
	
	print("The numbers classified as high are: ", highlist) #6c
		3

10	allnumbers = 0 #7
	
	for count in range(total):
	number = int(input('Enter a positive integer: '))
	
	while number < 1 or number > 100: #6b range check
	number = int(input('Out of range, please re-enter a positive integer: '))
	
	if number >= 1 and number <= 40:
	print(number, '- Low')
	elif number > 40 and number <= 70:
	print(number, '- Medium')
	elif number > 70 and number <= 100:
	print(number, '- High')
	highlist.append(number) #6c
	
	allnumbers = allnumbers + number #7
	
	print("The numbers classified as high are: ", highlist) #6c
	print("The average of all numbers are: ", allnumbers/count) #7
		3

Task 3

Question	Answer	Marks

11	Errors are underlined
	
	word = input("Please enter your word: ")
	word = word.islower()
	begin_p = word.beginswith("p")
	end_h = word.endswith("h")
	contain_e = word("e")
	word_length = word.length()
	if not begin_p and not end_h and contain_e = -1:
	  if word_length ≤ 3:
	  print("The length of the word is short.")
	  elif word <= 10:
	  print("The length of the word is medium.")
	  elif:
	  print("The length of the word is long.")
	if begin_p:
	  print("You entered a word that begins with 'p'.")
	elif end_h:
	  print("You entered a word that ends with 'h'.")
	elif not contain_e:
	  print("You entered a word that contains 'a'.")	

	Corrected Lines	
1	`word = word.lower()`	1
2	`begin_p = word.startswith("p")`	1
3	`contain_e = "e" in word`	1
4	`word_length = len(word)`	1
5	`if not begin_p and not end_h and not contain_e:`	1
6	`    if word_length <= 3:`	1
7	`    elif word_length <= 10:`	1
8	`    else:`	1
9	`elif contain_e:`	1
10	`    print("You entered a word that contains 'e'.")`	1

Task 4

Question	Answer	Marks
12	Create a conversion table from '0' to 'F'	1
	Use of a loop	2
	Correct calculation using % and //	
	Return hexadecimal number	1

Question	Answer	Marks
13	Create a signature line with correct function name and parameters *def randomnumber(start,end):*	1
	Correct use of randint with start and end values	1

Question	Answer	Marks
14	Create a signature line with correct function name and parameters *def createIPv4():* Return statement	1
	Call function randomnumber 4 times Concatenation of 4 numbers with '.'	1

Question	Answer	Marks
15	Create a signature line with correct function name and parameters *def storeIPv4(n):* Return statement	1
	Call function createIPv4 n times by loop	1
	Store all IPv4 addresses into a list	1

Question	Answer	Marks
16	Create a signature line with correct function name and parameters *def createIPv6():* Return statement	1
	Use of loop / multiple statements to generate 8 segments	1
	Call functions denary_to_hex and randomnumber	1

Question	Answer	Marks
17	Use of loop / multiple statements to generate 6 segments	1
	Call functions denary_to_hex and randomnumber	1

Question	Answer	Marks
18	Print 3 options on 3 lines.	1
	Validate input option as 1 or 2 or 3. Prompt for re-input if not 1 or 2 or 3.	1
	Correct conditional statements	1
	Calls all 3 functions	1

Sample Code:

```
#TASK 4 HEXADECIMAL.PY

#9 – 4M
def denary_to_hex(number):
    conversion_table = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A' , 'B', 'C', 'D', 'E', 'F']
    hexadecimal = ''

    while number>0:
        remainder = number%16
        hexadecimal = conversion_table[remainder]+ hexadecimal
        number = number//16
    return hexadecimal
```

```
#TASK 4 RANDOM.PY

#10 – 2M
def randomnumber(start, end):
    return random.randint(start, end)
```

```
#TASK 4 IPV4.PY

#11 – 2M
def createIPv4():
    ipv4 = str(randomnumber(0,255)) + "." + str(randomnumber(0,255)) + "." +
str(randomnumber(0,255)) + "." + str(randomnumber(0,255))
    return ipv4
```

```
#TASK 4 STORE.PY

#12 – 3M
def storeIPv4(n):
    ipaddr = []
    for x in range(n):
        ipaddr.append(createIPv4())
    return ipaddr
```

```
#TASK 4 IPV6.PY

#13 – 3M
def createIPv6():

    #IPv6 has 8 segments of 2 bytes
    segments = []
    for x in range(8):
        segments.append(denary_to_hex(randomnumber(0,65535))) #generate 8 segments

    IPv6 = str(segments[0]) + ":" + str(segments[1]) + ":" + str(segments[2]) + ":" + \
           str(segments[3]) + ":" + str(segments[4]) + ":" + str(segments[5]) + ":" + \
           str(segments[6]) + ":" + str(segments[7])

    return IPv6
```

```
#TASK 4 MAC.PY

#13 – 2M
```

```
def createMAC():

    #MAC has 6 segments of 1 byte
    segments = []
    for x in range(6):
        segments.append(denary_to_hex(randomnumber(0,255))) #generate 8 segments

    mac = str(segments[0]) + ":" + str(segments[1]) + ":" + str(segments[2]) + ":" + \
          str(segments[3]) + ":" + str(segments[4]) + ":" + str(segments[5])

    return mac
```

```
#TASK 4 NETWORK.PY

#14 – 4M
print("Option 1: Create an IPv4 address")
print("Option 2: Create an IPv6 address")
print("Option 3: Create a MAC address")

option = int(input("Please select options 1/2/3: "))

while option not in [1,2,3]:
    option = int(input("Re-enter options 1/2/3: "))

if option == 1:
    print(createIPv4())
elif option == 2:
    print(createIPv6())
else:
    print(createMAC())
```

-End of Paper-