Name:	

)

Class: \_\_\_\_\_



## MONTFORT SECONDARY SCHOOL

\_ (

## **PRELIMINARY EXAMINATION 2019**

Secondary 4 Express

COMPUTING Paper 2 (Lab-based)

11.30 am

Additional Materials:

7155/02 2 Sep 2019 (Mon)

2 hours 30 minutes

Electronic version of VEHICLES.xlxs data file Electronic version of VOWELS.py file Electronic version of SORTNUMBER.py file Insert Quick Reference Glossary attached as ANNEX A

# **READ THESE INSTRUCTIONS FIRST**

# Do not open this booklet until you are told to do so.

Write your name, index number and class in the spaces provided at the top of this page. Write in dark blue or black pen.

Answer all questions.

All tasks must be done in the computer laboratory. You are not allowed to bring in or take out any pieces of work or materials on paper or electronic media or in any other form.

Programs are to be written in Python. Save your work using the file name given in the question as and when necessary.

The number of marks is given in brackets [] at the end of each question or part question.

The total marks for this paper is 50.

For Examiner's Use			
Total	50		

Setter: Mr Wong Teck Piaw, Mr Ricky Tan **Task 1** 

A company uses spreadsheet software to record the list of vehicles they are interested in purchasing. You are required to finish setting up the spreadsheet to record the loan amount and monthly installment for each vehicle.

Open the file VEHICLES.xlsx. You will see the following data.

Save the file as MYVEHICLES\_<your name>\_<your class>\_<index number>.xlsx

1	A	В	C	D	E	F	G	Н	1
1 Vehicles List									
2									
3	Vehicle Model	Price	Downpayment	Loan Amount	Loan Type	Interest Rate (p.a.)	Loan Term	Monthly Instalment	Shortlist
4	BrandA 1.6A	\$67,988.00	\$47,591.60		Bank A (New vehicle)		7		
5	BrandA 1.6X	\$68,199.00	\$47,739.30		Bank A (Used vehicle)		7		
6	BrandB 1.6A	\$68,858.00	\$48,200.60		Bank B (New vehicle)		7		
7	BrandC 1.6A	\$67,876.00	\$47,513.20		Bank B (Used vehicle)		7		
8	BrandA 1.6A	\$68,419.00	\$47,893.30		Bank A (New vehicle)		7		
9	BrandB 1.6S	\$68,364.00	\$47,854.80		Bank B (Used vehicle)		7		
10	BrandA 1.6A	\$68,115.00	\$47,680.50		Bank B (Used vehicle)		7		e
11	BrandD 1.6S	\$68,609.00	\$48,026.30		Bank A (New vehicle)		7		
12	BrandB 1.6A	\$67,937.00	\$47,555.90		Bank A (Used vehicle)		7		
13	BrandC 1.6X	\$67,889.00	\$47,522.30		Bank B (New vehicle)		7		
14	BrandA 1.6A	\$68,490.00	\$47,943.00		Bank B (Used vehicle)		7		
15	BrandB 1.6B	\$68,729.00	\$48,110.30		Bank A (New vehicle)		7		
16	BrandD 1.6A	\$68,896.00	\$48,227.20		Bank B (New vehicle)		7		
17	BrandA 1.6A	\$68,578.00	\$48,004.60		Bank B (Used vehicle)		7		
18	BrandB 1.6S	\$68,876.00	\$48,213.20		Bank A (New vehicle)		7		e
19		_							
20	No. of vehicles								
21	Price range				Rates				
22					Description	Interest Rate per Year			
23					Bank A (New vehicle)	2.78%			
24					Bank A (Used vehicle)	2.99%			
25					Bank B (New vehicle)	2.79%			
26					Bank B (Used vehicle)	2.98%			

- 1 In cell **B20** enter a formula to count the number of vehicles listed.
- 2 In cell **B21** enter a formula to calculate the range of prices of vehicles listed.
- **3** Enter a formula to calculate the loan amount for each vehicle and use it to [1] complete the **Loan Amount** column. The loan amount is the balance after the downpayment is paid.
- 4 Use an appropriate function to search for the **Interest Rate per Year** in the **Rates** [2] table and use it to complete the **Interest Rate (p.a.)** column.
- **5** Enter a formula to calculate the monthly instalment for each vehicle and use it to [2] complete the **Monthly Instalment** column.

[1]

[1]

6 The company now decides to consider all BrandA vehicles only. Use a conditional statement to identify all BrandA vehicles and put Yes in the Shortlist column. Otherwise put No in the Shortlist column.

[3]

Save and close your file. Task 2

The alphabet is made up of 26 letters, 5 of which are vowels (a, e, i, o, u) and the rest of which are consonants. The following program accepts the input string 'computing' and prints out the number of occurrences for vowels in the string, and the consonants in the string.

```
vowel_num = 0
consonants = ''
for letter in 'computing':
    if letter in ['a','e','i','o','u']:
        vowel_num += 1
    else:
        consonants += letter
print('Number of Vowels: ',vowel_num)
print('Consonants: ',consonants)
```

#### Open the file VOWELS.py

Save the file as **MYVOWELS**\_<your name>\_<your class>\_<index number>.py

7 Edit the program so that it:

- (a) Also accepts vowels in capital letters (i.e. 'A', 'E', 'I', 'O' and 'U'). [1]
- (b) Asks the user for a string input and processes it instead of using the string [2] 'computing'.
- (c) Only allows string with length of less than 10 characters. Otherwise, it [3] displays the text "String must be less than 10 characters" and prompts for the string input again.
- (d) Displays the length of the consonants at the last line of the program output. [1]

Save your program.

8 Save your program as VARVOWELS\_<your name>\_<your class>\_<index number>.py

Edit your program so that it works for any number of string inputs. [3]

Save your program.



## Task 3

The following program should read integers from the user one at a time until a "q" is entered. Once all of the integers have been read, the program should display:

- All of the negative integers, followed by all of the zeros, followed by all of the positive integers.
- The numbers should be printed in the same order that they were entered by the user.
- Each number should be printed on its own line.

There are several syntax errors and logical errors in the program.

```
negative = [ ]
positives = [ ]
zeros = [0]
number = input("Enter an integer(enter q to quit): )
while number =! "q":
    num = str(number)
    if num < 0:
        negatives.append(num)
    elif num >= 0:
        positives.append(number)
    else
        zeros = zeros + 1
    number = input ("Enter an integer (enter q to quit): ")
print("The numbers were: ")
for n in negatives:
   print(n)
for z in zeros:
   print(z)
for p in positives:
    print(positives[p])
```

### Open the file SORTNUMBER.py

Save the file as SORTNUMBER \_<your name>\_<class>\_<index number>.py

9 Identify and correct the errors in the program so that it works correctly according to the rules above. [10]

Save your program.

## Task 4

You have been asked to write a program to classify and calculate the areas of 6 triangles.

The program should allow you to:

- Enter data in the format a,b,c where a, b, c are lengths of the three sides of a triangle An example is 11, 15, 20.
- Only allow data entry of integers in the range 10-99 (inclusive).
- Perform data validation to ensure that the Triangle Inequality Theorem holds: The Triangle Inequality Theorem states that the sum of the lengths of any 2 sides of a triangle must be greater than the length of the third side.
- Repeat this for a total of six triangles.
- Classify each triangle, and calculate the corresponding area rounded to 2 decimal places.
- The classification for the types of triangles is given below:

Classification	Characteristics
Equilateral	all sides are of equal lengths
Isosceles	only 2 sides are of equal lengths
Right Angled	satisfies Pythagorean Theorem. i.e $a^2 + b^2 = c^2$
Scalene	all 3 sides are of different lengths

• The Heron's formula to find the area of a triangle given the lengths of all sides is:

Area = 
$$\sqrt{s(s-a)(s-b)(s-c)}$$
, where s is  $s = \frac{a+b+2}{2}$ 

• Display this on the screen. Your output **must** look like this:

11,15,20	Scalene	81.39
30,40,50	Right Angled	600.0
33,66,66	Isosceles	1054.42
46,30,21	Scalene	248.37
27,27,27	Equilateral	315.67
24,26,10	Right Angled	120.0

**10** Write your program and test that it works.

Save your program as **TRI\_**<your name>\_<your class>\_<index number>.py [10]

[5]

- **11** When your program is working, use the following test data to show your test results:
  - 11,15,20 30,40,50 33,66,66 46,30,21 27,27,27 24,26,10

Take a screen shot of your results and save it as TRIRESULTS\_<your name>\_<your class>\_<index number>

Save your file in either .png or .jpg format.

### 12 Save your program as TRIAREAS <your name> <your class> <index number>.py

Extend your program to:

- Identify the largest triangle and add the label "(Largest)" beside the output.
- Calculate the average area of the triangles rounded to 2 decimal places, and output the average area at the last line of the program.

Your output should look like this:

11,15,20	Scalene	81.39	
30,40,50	Right Angled	600.0	
33,66,66	Isosceles	1054.42	(Largest)
46,30,21	Scalene	248.37	
27,27,27	Equilateral	315.67	
24,26,10	Right Angled	120.0	
Average a	area: 403.31		

Save your program.

13 Save your program as VARTRI\_<your name>\_<your class>\_<index number>.py

Extend your program to work for any number of triangles.

[2]

[3]

Save your program.

**End of Paper**