Class	Index Number	Name



# 新加坡海星中学 MARIS STELLA HIGH SCHOOL PRELIMINARY EXAMINATION SECONDARY FOUR

# COMPUTING

Paper 2 Practical (Lab-based)

**7155/02 29 Aug 2018** 2 hours 30 minutes

Additional Materials: Electronic version of TUITION.XLSX data file Electronic version of SORTNUMBER.PY data file Electronic version of HEIGHT.PY data file Insert Quick Reference Glossary

### READ THESE INSTRUCTIONS FIRST

Answer all the 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 number of marks for this paper is 50.



This document consists of 6 printed pages.

A&J Tuition Agency uses spreadsheet software to record the financial details of the company. You are required to finish setting up the spreadsheet to record the details.

Open the file **TUITION.XLSX**. You will see the following data.

Save the file as **TUITION** <your name> <class> <index number>

	Α	В	С	D	E	F	G
1	Salary Payout Table						
2							
			Working Hours				
З	Name	Job Title	(per Month)	Salary Scheme (per Hour)	Monthly Salary		
4	Alan Kim	Tutor	36				
5	Benedict Low	Office Staff	168				
6	Jasmine Heng	Office Front Staff	96				
7	Jeff Ong	Tutor	36				
8	Lester Lin	Office Front Staff	100				
9	Henry Wee	Tutor	36				
10	Mary Chan	Tutor	40				
11	Mabel Ong	Office Staff	160				
12	Jeffery Tan	Tutor	56				
13	Jerwellyn En	Tutor	56				
14				Total payout			
15							
16							
17						Job Schem	e Table
18						Job Title	Salary Scheme
19	Total hours of	of Tuition				Tutor	40
20	Tutor					Office Staff	25
21						Office Front Staff	17
22							
	Salary	Pay-out Table	Bank Loan Ove	erall (+)			: 4

Click on the Salary Pay-out Table worksheet to do the following questions.

- 1 Use an appropriate function to search for the **Salary Scheme** in the **Job Scheme Table** and use it to complete the **Salary Scheme** (per Hour) column.
- 2 Enter a formula to calculate the monthly salary for each worker and use it to complete the **Monthly Salary** column.
  - [1]

[2]

3 In cell **E14** enter a formula to calculate the total monthly salary paid out by the agency.

[1]

4 In cell **B20** enter a formula to calculate the total number of hours of tuition taught by the tutors of the agency.

[1]

Click on the **Bank Loan** worksheet to do the following questions.

5 In cell **B6** enter a formula to calculate the amount of loan repayment that is required based on the values in cell **B3** to **B5**.

Click on the **Overall** worksheet to do the following questions.

6 In cell **E7** enter a formula to calculate the amount of earnings of the agency, from tuition, based on the total number of classes, students per class and amount paid per student.

[1]

[1]

- 7 In cell **B10** enter a formula to calculate the total earnings of the tuition agency after deduction of the monthly liabilities.
- 8 In cell **B11** enter a conditional statement to identify the total earnings calculated in cell **B10**. If the total earnings calculated is below \$0, put **Loss**. If it is above \$0, put **Profit**. And if it is equal to \$0, put **Break Even**.

[1]

[1]

[3]

Save the file as **TUITION\_IF** <your name> <class> <index number>

**9** Use an appropriate feature of spreadsheet to find out how many students per class is required in order for the Total Profits per Month to be at "Break Even"

Save and close your file.

### Task 2

The following program accepts the height in metres(m) for 12 students and prints out the tallest height and the average height. The heights are in the range of 1.2m to 2.85m.

highest = 1.2 total\_height = 0 no\_of\_students = 12

for student in range (no\_of\_students):
 height = float(input("Enter the height of student in metres(m): "))
 if highest < height:
 highest = height
 total\_height = total\_height + height
 average height = total\_height/no of students</pre>

print("Highest height of student is ", highest) print("Average height of students is ", average\_height)

Open the file **HEIGHT.py** 

Save the file as MYHEIGHT\_<your name>\_<class>\_<index number>

**10** Edit the program so that it:

(a)	Accepts height for 36 students.	[1	]	

- (b) Prints out the shortest height as well as the tallest height. [4]
- (c) Test if the height input is between 1.2m and 2.85m (inclusive), and if not, asks the user for input again as necessary.

Save your program.

11 Save your program as CLASSHEIGHT\_<your name>\_<class>\_<index number>

Edit your program so that it works for any number of students.

[2]

[10]

Save your program.

### Task 3

The following program should read integers from the user 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 = ""
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(num)
else
    zeros += num
    number = input("Enter an integer(enter q to quit): )"
print("The numbers were: ")
```

```
for n in negatives:

print(negatives[n])

for z in zeros:

print(zeros[z])

for p in positives:

print(positives[p])
```

### Open the file SORTNUMBER.py

Save the file as MYNUMBER\_<your name>\_<class>\_<index number>

**12** Identify and correct the errors in the program so that it works correctly according to the rules above.

Save your program.

### Task 4

You have been asked to write a program to verify the check digit of an ISBN number. The last number at the end of the ISBN number is called a check digit. The check digit allows us to check if the ISBN number has been entered correctly.

The algorithm for generating the check code is as follows:

 Obtain the weighted sum of the ISBN digits using the weights [1, 3, 1, 3, 1, 3, 1, 3, 1, 3, 1, 3].
 For ISBN number 978-0-306-40615-7, the sum is

 $(9 \times 1) + (7 \times 3) + (8 \times 1) + (0 \times 3) + (3 \times 1) + (0 \times 3) + (6 \times 1) + (4 \times 3) + (0 \times 1) + (6 \times 3) + (1 \times 1) + (5 \times 3) = 93$ 

- Find the remainder of the weighted sum when divided by 10.
- If the remainder is equal to 0 then the check digit equals to zero
- If not, subtract the remainder from 10 to obtain the check digit.

The check digit 7 for 978-0-306-40615-7 is correct.

The program should allow you to:

- Enter an ISBN number in the format 000-0-000-00000-0
- Ensure that only numbers in the correct format is accepted
- Calculate the weighted sum of the ISBN number.
- Calculate the remainder of the weighted sum.
- Calculate the check digit and verify if the ISBN entered is correct.
- Display the output on the screen. Your output **must** look like this:

The ISBN no. to check is :	123-3-333-43234-0
Weighted sum is :	70
Calculated check digit:	0
The ISBN number is correct.	

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

[10]

Save your program as MYISBN\_<your name>\_<class>\_<index number>

**14** When your program is working, use the following test data to show your test results:

123-3-333-43234-0 425-9-594-49137-3 784-2-362-74829-8

Take a screen shot of your results and save it as a bitmapISBNRESULT\_<your name>\_<class>\_<index number>[5]

15 Save your program as **ISBNCON**\_<your name>\_<class>\_<index number> Extend your program to continuously check the numbers until "q" is entered

Save your program.

**16** Save your program as **ISBNCHANGE**\_<your name>\_<class>\_<index number> Extend your program to allow the user to enter the weights and calculate the check digits based on the new weights.

The user should be asked to key in weights in the format [x,x,x,x,x,x,x,x,x,x,x,x,x] (should only be 12 digits) then ask for the ISBN number to be checked.

Save your program

-End of Paper-

[3]