Thumb drive No:	

Class/ Index Number	Centre Number/ 'O' Level Index Number	Name
/	/	

新加坡海星中学

MARIS STELLA HIGH SCHOOL
PRELIMINARY EXAMINATIONS
SECONDARY FOUR

COMPUTING **7155/02**
Paper 2 Practical (Lab-based) **28 August 2020**
2 hours 30 mins

Additional Materials:
Electronic version of COMPANY.XLSX file
Electronic version of DEPOSITS.PY file
Electronic version of CASHBACK.PY file
Insert Quick Reference for Python

READ THESE INSTRUCTIONS FIRST

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 number of marks for this paper is 50.

For Examiner's Use
50

This document consists of **6** printed pages.

Task 1

A maintenance company uses spreadsheet software to record details of its completed jobs. There are five different job categories provided by the company. For each type of job, the total cost comprises a deposit amount and hourly cost.

You are required to finish setting up the spreadsheet to calculate the total cost of the jobs.

Open the file **COMPANY.xlsx**. You will see the following data.

	A	B	C	D	E	F	G	H	I	J
1	**Spick and Span Pte Ltd**									
2	**Job Details**									
3	**No.**	**Job type**	**Job Category**	**Job ID**	**Job Code**	**Deposit**	**Rate Per Hour**	**Hours Required**	**Hourly Cost**	**Total Cost**
4	1	Aircon Cleaning	Home Maintenance	PQ001235-HM			$50.00	3		
5	2	Carpentry	Interior Design	PQ003234-ID			$60.00	6		
6	3	Area Cleaning	Home Maintenance	PQ001236-HM			$20.00	2		
7	4	Electrical Cabling	Electrical Engineering	PQ002234-EE			$40.00	2		
8	5	Engineering Design	Mechanical Engineering	PQ002235-ME			$38.00	4		
9	6	Flooring	Interior Design	PQ003234-ID			$65.00	5		
10	7	Gardening	Horticulture	PQ005234-HC			$25.00	2		
11	8	Mowing	Home Maintenance	PQ001237-HM			$23.00	5		
12	9	Plumbing	Home Maintenance	PQ001234-HM			$70.00	2		
13					**2nd Highest Hourly Rate**			**TOTAL EARNED**		
14										
15										
16		**Job Code Fees**								
17		**Job Code**	**Deposit**							
18		HM	$120.00							
19		EE	$180.00							
20		ID	$150.00							
21		HC	$120.00							
22		ME	$175.00							
23										

Save the file as **JOBS_<class>_<index number>_<your name>.xlsx**

1 In cells **E4** to **E12**, enter a formula that uses an appropriate function to extract the last 2 letters of the corresponding **Job ID.** The Job ID is a 11 character text. [2]

2 In cells **F4** to **F12**, enter a formula that uses an appropriate function to search for the **Deposit** in the **Job Code Fees** table. [2]

3 In cells **I4** to **I12**, enter a formula to calculate the **Hourly Cost**, which is the product of **Rate Per Hour** by **Hours Required**. In cells **J4** to **J12**, enter a formula to calculate **Total Cost**, which is the **Deposit** added to the **Hourly Cost**. [2]

4 Format the cells **J4** to **J12** to automatically highlight those cells whose total cost is more than **$400** with a **red** background. [1]

5 In cells **J13**, enter an appropriate function to calculate the total earned. [1]

6 In cell **G13**, enter an appropriate function to find the second highest hourly rate. [2]

Save and close your file.

Task 2

A bank stores the names and deposits of its customers into two lists. The following program consists of five customer names and their corresponding deposits. It prints the names and deposits of the customers whose name begins with the letter 'J'.

```
names = ['Joy', 'Mary', 'John', 'Dean', 'Siva']
deposits = [540.35, 525.60, 539.50, 500.05, 570.95]

for i in range(len(names)):
    if names[i].startswith('J'):
        print("Customer name is {}, Deposit amount is {}".format(names[i], deposits[i]))
```

Open the file **DEPOSITS.py**

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

7 Edit the program so that it prints out the names and deposits of the customers whose names begin with the letter 'D'. Do not print those beginning with the letter 'J'. [1]

8 The program will then calculate the total deposit and average deposit of these five accounts.

(a) Edit the program to: [2]

- calculate the total amount of deposit
- output the message "The total deposit is: ", with the total amount of deposit.

Save your program.

(b) Edit the program to: [3]

- calculate the average deposit rounded to the nearest whole number.
- output the message "The average deposit amount is: ", with the average deposit amount.

Save your program.

(c) Edit the program to display the name of customers whose names end with the letter 'n' and length of name is equal to 4. [4]

Save your program.

Task 3

The following program calculates the amount of cashback discount that a customer receives based on the following rules applied to the amount of expenditure in a quarter:

- A customer receives a 5 percent cashback for expenditure of $500 or more, or total accumulated rewards are 1000 points or more
- A customer receives a 3 percent cashback for expenditure of $300 or more, and total accumulated rewards are 700 points or more
- A customer receives a 2 percent cashback for expenditure of $200 or more

The expenditures and rewards are whole numbers. The program ends when the user enters 'N' for no further customer input.

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

```
customers = False
count = 10

while customers == True:
    exp = int(input("Enter expenditure for quarter: "))
    rewards = int(input("Enter rewards accumulated: "))

    if exp >= 500 and rewards >= 1000:
        print("Customer receives 5 percent cashback.")
        count = count += 1
    elif exp > 300 and rewards >= 700:
        print("Customer receives 3 percent cashback.")
    elif exp >= 200:
        print("Customer receives 2 percent cashback.")
    else
        print("No cashback discount."")

    more_cust = int(input("Any more customers? Type Y or N: "))

    if more_cust == 'Y':
        customers = False
    else:
        customers = False

print("Total customers who qualify for 5 percent cashback are: " + count)
```

9 Open the file **CASHBACK.py** [10]

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

Task 4

You have been asked to create a word guessing game. A user enters a word of six or fewer letters for guessing. The user has six chances to guess all letters of the word.

Assume that all inputs and outputs are in lowercase letters. There is no need to validate for case.

The program must:

- Output at the start of the game "You have 6 chances to guess a word."
- Allow user to input a word with six or fewer letters for guessing. There must be validation to check that the input word does not have more than six letters.
- Create and output a new list with length equal to that of the word. This list will be updated later if letters are found. For a six letter word like 'banana', the list looks like `['-', '-', '-', '-', '-', '-']`.
- Allow user to have up to six chances to input a letter at a time. Validation of the letter is not needed. Assume that the input is a single lowercase letter.
- Check if the input letter is found in the word. If the letter is found, output "Letter found in word." If the letter is not found, output "Letter not found."

10 Write your program and test that it works. [10]

Save your program as **WORDGUESS1_<class>_<index number>_<your name>.py**

11 When your program is complete, test it for the following: [2]

- Test 1 – Input the word 'banana', then letters 'g', 'o', 'b', 'a', 'n' in that order.
- Test 2 – Input the word 'mangosteen', then word 'mango', then letters 'm', 'a', 'n', 'g', 'o' in that order

Take a screenshot of:
- Test 1. Save this screenshot as **TEST1_<class>_<index number>_<your name>**
- Test 2. Save this screenshot as **TEST2_<class>_<index number>_<your name>**

Save your files either in **.png** or **.jpg** format.

12 Save your program as **WORDGUESS2_<class>_<index number>_<your name>.py**. Extend your program to: [8]

- Create a new list to store the index(es) of all occurrences of the letter if found in the word. Using the index(es), update the first list to output the letters that have been guessed correctly. For a six letter word like 'banana' and the input letter is 'a', the list looks like `['-', 'a', '-', 'a', '-', 'a']`.
- If all six chances were used up, output "Game over. All chances used!".

- If all letters have been correctly guessed, output “You win!” and the program must end.

-End of Paper-