ZHONGHUA SECONDARY SCHOOL

PRELIMINARY EXAMINATION 2020
SECONDARY 4 EXPRESS

Candidate's Name	Class	Register Number

COMPUTING

7155/02

Paper 2 (Lab Based)

16 September 2020
2 hours 30 minutes

Additional Materials: Instruction Set, Thumb Drive

READ THESE INSTRUCTIONS FIRST

Write your index number and name on all the work you hand in.
Write in dark blue or black pen on both sides of the paper.
You may use a pencil for any diagrams or graphs.
Do not use staples, paper clips, glue or correction fluid.

Answer **all** the questions.
Write your answers in this question booklet.
Give non-exact numerical answers correct to 3 significant figures, or 1 decimal place in the case of angles in degrees, unless a different level of accuracy is specified in the question.
The use of a scientific calculator is expected, where appropriate.
You are reminded of the need for clear presentation in your answers.

At the end of the presentation, fasten all your work securely together.
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:

Setter: Mr. Calvin Heng
Vetter: Mr. Low Kee Ley

Task 1

Good Health Services is a health club offering spa and fitness related equipment and services to both members and day guests. The health club has five types of memberships and offers different rebates for members. You are required to finish setting up the spreadsheet to calculate the cost or each customer.

Open the file **GOODHEALTH.xlsx**

	A	B	C	D	E	F	G
1	**Good Health Services Payments for Month of December**						
2							
3	**First Name**	**Last Name**	**Number of Days**	**Member Type**	**Cost**	**Rebate**	**Total Cost**
4	Avril	Morgan	20	Bronze			
5	Anna	Wilson	25	Silver			
6	Beatrice	Smith	12	Platinum			
7	Brandon	Potter	13	Gold			
8	Charles	Manson	17	Day			
9	Cali	Li	8	Iridium			
10	Denver	Dobson	30	Bronze			
11	Diana	Williams	26	Silver			
12	Edward	Norton	17	Iridium			
13	Elise	French	5	Day			
14	Florence	Nightingale	20	Silver			
15	Favian	Kaisaros	14	Gold			
16	George	Wittman	13	Gold			
17	Gretta	Gable	25	Silver			
18	Helga	Herman	4	Iridium			
19							
20	Number of Customers				Average Revenue		
21	Number of Members						
22							
23							
24			**Membership Types Table**				
25		**Day**	**Bronze**	**Silver**	**Gold**	**Platinum**	**Iridium**
26	**Cost per Day**	45	50	75	95	105	145

Save the file as:
GOODHEALTH _<YOUR NAME>_<INDEX NO>

1 In cell C20 enter a formula to count the number of customers. [1]

2 In cell C21 enter a formula to count the number of members. [1]

3 Use an appropriate function to search for the **Cost** for each customer and use it to complete the **Cost** column. The cost must take into consideration the **Number of Days** each customer used the club. [2]

4 Use a conditional statement to calculate the rebate given for members only. Bronze receives 3% of the monthly cost, Silver receives 5% of the monthly cost, Gold receives 7% of the monthly cost, Platinum receives 13% of the monthly cost and Iridium receives 17% of the monthly cost. [2]

5 Calculate the Total Cost for each customer. [2]

Total Cost = Cost Less Rebate

6 In cell **G20** enter a formula to calculate the average total cost, rounded to the nearest dollar($). [2]

Save and close your file.

Task 2

Every morning, students of a class (10 students) are required to report their temperature reading. The following program allows the teacher to input the temperature reading for each student (identified by Student Index Number).

Open the file **TEMPERATURE_READINGS.py**

```
# Declare variables
temp = 0        # to store temperature value
index = 0       # to store student index Number
student = []    # to store index and temperature of individual student
class+data = [] # to store student details of index and temperature reading for that class
class_size = 10 # number of readings to take

# Input
for count in range(class_size):
    index = input("Enter student index number: ")
    temp = int(input("Enter the temperature: ")
    student = [index, temp]
    class_data.append(student)

# Processing




# Output
```

Save the file as:
TEMPERATURE_READINGS_<YOUR NAME>_<INDEX NO>.py

7 Edit the program so that it:

(a) Accepts 20 students, as more students join the class. [1]

(b) Print out the students with temperature > 37.9 degrees Celsius. [2]

(c) Test if the temperature reading is between 35 and 40 degrees (inclusive) during input. If not, ask the user for input again as necessary. [3]

Save your program.

8 Save your program as:
Override _<YOUR_NAME> _<INDEX_NO>

9 Edit your program so that it will allow any number of student temperature readings to be taken. At the end of the temperature input, the program will also print out the Average Temperature Reading for that session. [4]

Save your program.

Task 3

The following program is a Number Base Converter. It converts an eight bit binary number to its equivalent denary number.

The program follows these rules:

- The 8 bit binary number is entered as a string.
- If the first bit (from the right) is a '1', the answer is accumulated with 2^0. If the second bit (from the right) is a '1', the answer is accumulated with 2^1. This is done until the last bit (27).
- The end result of the accumulations is the number in denary.

The program prints out the denary value.

There are several syntax and logic errors in the program.

Line	Program Code
001	# Binary (8 bit) to Denary Converter
002	number_of_bits = 7
003	
004	# Get user to input binary string
005	binary_string = int(input('Please enter the 8 bit binary number: '))
006	
007	# Calculate the denary number
008	answer = 0
009	for i in range(number_of_bits):
010	if binary[i] == '0':
011	answer += answer * 2*(number_of_bits - i)
012	
013	# Display the denary number
014	print("The denary number is + answer)

Open the file **B_To_D.py**

Save the file as:
B_To_D_<YOUR NAME>_<INDEX NO>.py

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

Save your program.

Task 4

You have been asked to create a Bit Checker program.

The program should:

- Allow the user to enter a 7 bit data string of binary numbers.
- Your program then counts the number of '1's in the data string.
- If the number of '1's is ODD, a zero '0' is appended to the right of the data string.
- If the number of '1's is EVEN, a one '1' is appended to the right of the data string.
- You are to choose a suitable data structure to store this newly formed data string.
- Your program is to allow for 7 data strings to be stored in the data structure.

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

Save your program as
Bit_Checker_<YOUR_NAME>_<INDEX_NUMBER>.py

12 When your program is complete, test it for the following:

- 1000100 will result in 10001001
- 0111011 will result in 01110110
- 1100110 will result in 11001101
- 0011001 will result in 00110010
- 1011011 will result in 10110110
- 0100001 will result in 01000011
- 0101010 will result in 01010100 [3]

Take a screenshot of the data structure contents. Name your screenshot as:
TEST1_<YOUR_NAME>_<INDEX_NUMBER> [1]

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

13 Save your program as
Bit_Checker2D_<YOUR_NAME>_<INDEX_NUMBER>.py

Your data structure will now accept an 8th data string of 8 bits. You are to construct these 8 bits as follows: (please refer to the test data set in #12 above)
- Count the number of '1's in the first column. If the answer is ODD, then place a zero '0' in the first position (from the left). If the answer is EVEN, then place a one '1' in the first position (from the left).
- Count the number of '1's in the second column. If the answer is ODD, then place a zero '0' in the second position (from the left). If

the answer is EVEN, then place a one '1' in the second position (from the left).
- Continue until you finish all the columns.

The example test data in #12 will yield a bit string of 01011110 [1]

Take a screenshot of the data structure contents. Name your screenshot as:
TEST2_<YOUR NAME>_<INDEX_NUMBER> [1]

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

Save your program.

14 Save your program as
Bit_Checker_Detect_<YOUR_NAME>_<INDEX_NUMBER>.py

Extend your program such that if any one bit in the first seven bit strings is flipped (i.e., changed from zero to one or from one to zero), your program has the ability to detect and print out the faulty bit string, including a corrected bit string.

[2]

Take a screenshot of the data structure contents (Flip a bit of your choice). Name your screenshot as:
TEST3_<YOUR NAME>_<INDEX_NUMBER> [1]

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

Take a screenshot of the detected bit string with correction. Name your screenshot as:
TEST4_<YOUR NAME>_<INDEX_NUMBER> [1]

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

Save your program.

End of Paper