Name:	(

Class: _____



MONTFORT SECONDARY SCHOOL

)

PRELIMINARY EXAMINATION 2020

Secondary 4 Express

COMPUTING Paper 2 Practical (Lab-based)

7155/02 1 Sep 2020 (Tue)

1.00 pm

2 hours 30 minutes

Additional Materials:

Electronic version of LOANS.xlxs data file Electronic version of NUMBER.py file Electronic version of NAPFA.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

This document consists of 8 printed page.

Parent's Signature: _____

Setter: Mr Ricky Tan

A loan broker keeps track of his clients' housing loans using a spreadsheet. The rates of three banks are compared and the best (lowest) rate is used to calculate each client's Monthly Payment.

You are required to finish setting up the spreadsheet.

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

	А	В	С	D	E	F	G	Н	I	J
	Loan Number	Principal	Years	Rate	Rate	Rate	Best	Monthly	Total Payment	Total Interest
1				Α	В	С	Rate	payment	made	paid
2	1	\$80,000	10	3.1%	2.9%					
3	2	\$500,000	12	2.5%	2.5%					
4	3	\$300,000	10	2.5%	2.5%					
5	4	\$1,000,000	25	2.1%	2.3%					
6	5	\$1,800,000	25	1.5%	1.6%					
7	6	\$320,000	30	2.5%	2.5%					
8	7	\$640,000	22	2.5%	2.5%					
9	8	\$120,000	12	2.8%	2.7%					
10	9	\$980,000	20	2.5%	2.5%					
11	10	\$1,200,000	30	2.1%	2.3%					
12	11	\$1,590,000	25	1.5%	1.6%					
13	12	\$95,000	8	3.1%	2.9%					
14										
15	15 Rates									
16	Loan Amount	Rate	e (% p.a)							
17	(at least)	Bank A	Bank B	Bank C						
18	\$50,000	3.1%	2.9%	3.2%						
19	\$100,000	2.8%	2.7%	2.9%						
20	\$250,000	2.5%	2.5%	2.4%						
21	\$1,000,000	2.1%	2.3%	2.2%						
22	\$1,500,000	1.5%	1.6%	1.7%						
23	\$2,000,000	1.3%	1.4%	1.5%						

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

- 1 In cells **F2** to **F13** enter a formula that uses an appropriate function to search for the rate offered by Bank C for each given principal. The rates offered by each bank is given in the Rates table.
- In cells G2 to G13 enter a formula to determine the Best Rate among the rates offered by the three banks. The best rate is the lowest of Rate A, Rate B, and Rate [2] C
- 3 In cells H2 to H13 enter a formula to calculate the **Monthly Payment** of a loan based on the Best Rate.

[2]

[2]

- 4 In cells **I2** to **I13** enter a formula to calculate the **Total Payment made** of a loan based on the Best Rate. The Total Payment made is the sum of all monthly payments for the duration of the loan.
- 5 In cells **J2** to **J13** enter a formula to calculate the **Total Interest paid** of a loan based on the Best Rate. The Total Interest paid is amount of interest that was paid to the bank for the duration of the loan.

[2]

[2]

Task 2 begins on the next page.

The following program is a number guessing game that allows the user to guess a randomly generated number between 1 and 10 inclusive.

The program does the following:

- Generates a random number between 1 and 10 inclusive;
- Allows the user to repeatedly guess the randomly generated number by entering a number;
- Informs the user that his guess is correct if his guess matches the randomly generated number and the program ends;
- Otherwise, informs the user whether the correct number is greater than or less than his guess.

```
import random
```

```
ans = random.randint(1,10)
x = int(input("Enter a number: "))
while x != ans:
    if x < ans:
        print("Your guess is too low!")
    else:
        print("Your guess is too high!")
    x = int(input("Enter a number: "))
```

print("Correct!")

Open the file NUMBER.py

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

- 6 Edit the program so that it:
 - (a) Generates a number between 1 and 100, inclusive, instead of 1 to 10. [1]
 - (b) Counts the number of tries the user has used, and prints it out at the end. [3]
 - (c) Performs data validation on the user input. Informs the user if his input is not an integer and allow him to enter again. [4]

Save your program.

7 Save your program as MYNUMBER8_<your name>_<your class>_<index number>.py

Extend your program so that the user can only guess up to 8 times. Output an appropriate message if he has used up all 8 tries and has yet to correctly guess [2] the number.

Save your program.

The following program calculates the number of Points scored by a 16 year-old male student doing his Standing Broad Jump station in a NAPFA test. Subjects are given two tries and the better of the tries is used to determine the score he scores. The score table is shown below.

Points	Distance Jumped (cm)
5	> 245
4	236-245
3	226-235
2	216-225
1	206-215
0	< 206

The program does the following:

- Allows the PE teacher to enter the distances that a student jumped.
- Calculates the points that the student scored based on the better of the two distances.

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

```
try1 = int(input(Enter distance for Attempt 1: "))
try1 = int(input("Enter distance for Attempt 2: "))
if try1 < try2:
    dis = try1
elif:
   dis == try2
if dis >= 245:
   print("5 Points")
if dis >= 236:
   print("4 Points")
elif dis >= 226:
   print("3 Points"
elif dis >= 216:
   print("2 Points")
elif dis \geq 206:
   print("1 Points")
else
   print("6 Points")
```

Open the file NAPFA.py

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

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

Save your program.

In computer Role Playing Games (RPGs), a world is presented to the user and he explores it by moving about, picking up valuable items and fighting enemies along the way.

You have been asked to write a simple RPG.

A small world can be represented by a 4×4 grid, with each space indexed by a number.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

The game starts by presenting the initial state of the world to the user. Each space is represented by a character:

Character	Represents
	Unexplored space
A	Space explored by Player A

The player always starts at space 0, which is at the top-left corner of the world.

- **9** Represent the world in a suitable structure. Construct the initial state of the world and display it on the screen. Your output must look like this:
 - A....

Write your program and test that it works.

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

[4]

10 Movement in the world is done using the keys 'w', 's', 'a', 'd', which correspond to

Key	Movement
w	Up
S	Down
а	Left
d	Right

Extend your program to do the following:

- Asks the user to enter his move sequence as a string. For example, entering "sdsdsd" means to move in the sequence down, right, down, right, down, right.
- Output the updated state of the world to the user with all visited spaces represented by the letter 'A' instead of '.'. For example, entering "sdsdsd" will result in the updated state:
 - A... AA.. .AA. ..AA

- Note that:
 - You can assume that users will always enter move sequences that are within the given grid.
 - It is alright to visit a space more than once.

Write your program and test that it works.

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

11 Use the following test data to test your program:

ssddds

Take a screen shot of your results and save the screenshot as

GAMERESULT_<your name>_<your class>_<index number>

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

12 Save your program as

GAMEB_<your name>_<your class>_<index number>.py

Extend your program to a 2-player game by adding the following:

- Prompt for Player B's move sequence after Player A has entered his.
- Player B starts at space 15, which is the bottom-right corner of the world.
- Output the updated state incorporating both players' moves.
- Show 'B' in spaces visited by Player B only.
- Show 'x' in spaces visited by both Player A and B.

Save your program.

[7]

[2]

[5]

13 Save your program as

GAMEVAR_<your name>_<your class>_<index number>.py

Extend your program to work with square-shaped worlds of any size, adhering to the previous rules:

[2]

- Player A starts at the top-left corner of the world.
- Player B starts at the bottom-right corner of the world.

Save your program.

End of Paper