SCHOOL OF SCIENCE AND TECHNOLOGY, SINGAPORE

SECONDARY 4
PRELIM EXAMINATION

COMPUTING

Paper 2 **7155/02**

Practical (Lab-based)

14 August 2020 (Friday) **2 hours 30 minutes**

CANDIDATE NAME	

CLASS		INDEX NUMBER		

Additional Materials: Electronic version of `CAR.XLSX` data file
Electronic version of `PASSWORD.PY` file
Electronic version of `SHN.PY` file
Insert Quick Reference Glossary

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.

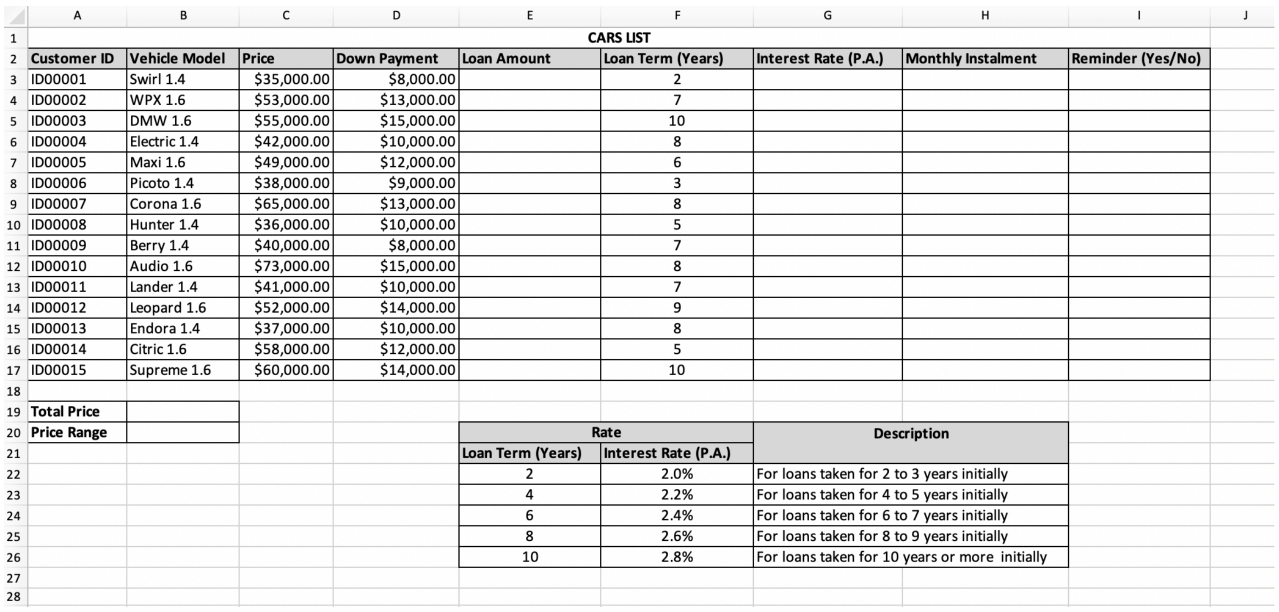
The number of marks is given in brackets [ ] at the end of each question or part question. The total number of marks for this paper is 50.

For Examiner's Use		
1	1	
2	1	
3	2	
4	2	
5	2	
6	2	
7	1	
8	2	
9	3	
10	4	
11	10	
12	12	
13	5	
14	3	
Total	/50	

Task 1

A car dealer uses a spreadsheet software to record the list of customers who recently bought a car from them, and the type of loans they made. You are required to finish setting up the spreadsheet. For consistency, you are also required to display all results as **positive numbers**.

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



	A	B	C	D	E	F	G	H	I	J
1						CARS LIST				
2	Customer ID	Vehicle Model	Price	Down Payment	Loan Amount	Loan Term (Years)	Interest Rate (P.A.)	Monthly Instalment	Reminder (Yes/No)	
3	ID00001	Swirl 1.4	$35,000.00	$8,000.00		2				
4	ID00002	WPX 1.6	$53,000.00	$13,000.00		7				
5	ID00003	DMW 1.6	$55,000.00	$15,000.00		10				
6	ID00004	Electric 1.4	$42,000.00	$10,000.00		8				
7	ID00005	Maxi 1.6	$49,000.00	$12,000.00		6				
8	ID00006	Picoto 1.4	$38,000.00	$9,000.00		3				
9	ID00007	Corona 1.6	$65,000.00	$13,000.00		8				
10	ID00008	Hunter 1.4	$36,000.00	$10,000.00		5				
11	ID00009	Berry 1.4	$40,000.00	$8,000.00		7				
12	ID00010	Audio 1.6	$73,000.00	$15,000.00		8				
13	ID00011	Lander 1.4	$41,000.00	$10,000.00		7				
14	ID00012	Leopard 1.6	$52,000.00	$14,000.00		9				
15	ID00013	Endora 1.4	$37,000.00	$10,000.00		8				
16	ID00014	Citric 1.6	$58,000.00	$12,000.00		5				
17	ID00015	Supreme 1.6	$60,000.00	$14,000.00		10				
18										
19	Total Price									
20	Price Range				Rate		Description			
21					Loan Term (Years)	Interest Rate (P.A.)				
22					2	2.0%	For loans taken for 2 to 3 years initially			
23					4	2.2%	For loans taken for 4 to 5 years initially			
24					6	2.4%	For loans taken for 6 to 7 years initially			
25					8	2.6%	For loans taken for 8 to 9 years initially			
26					10	2.8%	For loans taken for 10 years or more initially			
27										
28										

Save your file as **CAR_**<your name>_<index number>**.xlsx**

1. In cell B19, enter a formula to calculate the total price of the vehicles sold. **[1]**

2. In cell B20, enter a formula to calculate the range of prices of vehicles sold. **[1]**

3. The loan amount is the amount of money for the car not paid in the down payment. In cells E3 to E17, enter a formula that uses an appropriate function to calculate the loan amount each customer has to take to complete the **Loan Amount** column.
[2]

4. In cells G3 to G17, enter a formula that uses an appropriate function to search for the **Interest Rate Per Annum (P.A.)** in the **Rate** table and use it to complete the **Interest Rate (P.A.)** column. **[2]**

5. In cells H3 to H17, enter a formula to calculate the monthly instalment each customer has to pay and use it to complete the **Monthly Instalment** column. **[2]**

6. A reminder will be sent to customers who have taken more than or equal to $28 000 in loans and whose loan terms are 5 years or less. In cells I3 to I17, enter a conditional statement to identify such customers and put **Yes** in the **Reminder** column. Otherwise put **No** in the cell. **[2]**

Save and close your file.

[Turn over

Task 2

The following program simulates a scenario that a user sets a password for the first time. After that, the system allows the user to have three attempts in logging into the system with the correct password.

```
#User sets password and tries to log into a system

password = input("Please set password: ")
print("Your password is : ", password)

count = 3
hit = False
while count > 0 and hit == False:
    print("Please log into the system.")
    print("You have {} attempt(s) left.".format(count))
    user_input = input("Please enter the password: ")
    if user_input == password:
        print("You have successfully logged into the system.")
        hit = True
    else:
        print("You did not enter the correct password.")
        count -= 1

if count == 0:
    print("You are locked out of the system.")
```

Open the file **PASSWORD.py**
Save the file as **PASSWORD**_<your name>_<index number>**.py**

Edit the program so that it:

7. Allows up to 4 password attempts when logging into the system. **[1]**

8. Prints out the reversed form of the password (e.g. "ABCD" is printed as "DCBA") right after a valid password has been set. **[2]**

9. Ensures that the password consists of at least 8 characters. Otherwise, the program should inform the user about the input requirements and ask for re-entry of the input. **[3]**

10. Ensures that the set password consists of at least 1 upper case alphabet ('A' to 'Z'). Otherwise, the program should inform the user about the input requirements and ask for re-entry of the input. **[4]**

Task 3

The following program takes in a date, DS, as an input string with the format "DD MM YYYY". It then adds 14 days to DS and outputs the new date.

The program takes into consideration the leap years.

A leap year is exactly divisible by four except for years that are exactly divisible by 100. However, these centurial years are leap years if they are exactly divisible by 400.

There are several logical and syntax errors in the program.

```
#Input
DS = input("DD MM YYYY:").split()
MM, Yr = int(DS[1], DS[2])

#Check if it is Leap year
leap_yr = false
if (Yr % 4) == 0:
   if (Yr % 100) == 0:
       if (Yr % 400) == 0:
           leap_yr = True
else:  leap_yr = True

#Check for total days of the month
max_day = 31
if MM in [4, 6, 8, 11]:
    max_day = 30
elif MM = 2:
    max_day = 28

newDD, newMM, newYr = int(DS[0])+14, MM, Yr

#Check for day, month, year value overflow
if newDD > max_day:
    newDD = newDD MOD max_day
    newMM += 1
    if newMM > 12:
        newMM = 1
        newYr -= 1

#Output result
print("+14 days\nDD: {} MM: {} YYYY: ".format(newDD,newMM,newYr))
```

Open the file **SHN.py**
Save the file as **MYSHN**_<your name>_<index no>.**py**

11. Identify and correct the errors in the program

Save your program. **[10]**

[Turn over

Task 4

Due to the Circuit Breaker implementation, Keanu is unable to meet up regularly with his classmates to play their favourite tabletop role-playing game (TRPG). The TRPG uses six types of different sided dice for different occasions. There are four-sided (*d*4), six-sided (*d6*), eight-sided (*d*8), ten-sided (*d*10), twelve-sided (*d*12) and twenty-sided (*d*20) dice.

The dice rolling is split into two main types: a to-hit roll and a damage roll.

I. A to-hit roll is done using a single twenty-sided die plus a value *B*. If the roll is equal to or greater than a target number (*Tn*), then it is considered a success roll; otherwise it is a failed roll. If this roll is successful, a damage roll will be required.

II. A damage roll is done using *N* number of *X*-sided dice plus a value *C*. For example 2*d*6 +10, will mean rolling two six-sided dice, summing up the value and then adding 10. Such rolls can be generalised as *NdX* + *C*.

You have been asked to write a program to do the following:

- Output the message "`Enter values for B, Tn, N, X, and C, each separated by a space:`"
- Allow user to input a string with the following format "`B Tn N X C`" where
  - $0 \leq B, C \leq 10$
  - $5 \leq Tn \leq 30$
  - $1 \leq N \leq 20$
  - $X \in \{4, 6, 8, 10, 12, 20\}$
- Output "`Invalid input`" if any of *B*, *Tn*, *N*, *X* or *C* are not within the valid range, and ask for re-entry.
- Output the message "`1d20 + B, Target Number: Tn`", where *B* and *Tn* are the input values.
- Output a single integer, *P*, where *P* is the sum of *B* and a randomly generated value between 1 and 20.
- Output "`Fail`" if *P* is less than *Tn*, then end the program. Otherwise output "`Pass`".
- Output "`Damage: NdX + C`", where *N*, *X* and *C* are the input values.
- Output *N* numbers of randomly generated values between 1 to *X* in a single line.
- Output "`Total: S`", where *S* is the sum of the *N* random numbers plus *C*, then end the program.

Additional Notes: To randomly generate a number you can use `randint()` function from Python's random module.

```
import random
#This will generate a number between 1 and 6, inclusive.
print(random.randint(1,6))
```

12. Write your program and test that it works.
Save your program as **DICEROLLER1**_<your name>_<index number>**.py**

[12]

13. When your program is complete, test for the following:
Test 1 -- User inputs the string "4 15 0 6 10"
Test 2 -- User inputs the string "3 4 1 9 8"
Test 3 -- User inputs the string "0 10 4 4 12"
Test 4 -- User inputs the string "2 23 3 6 10"
Test 5 -- User inputs the string "10 11 15 10 3"

Take a screenshot of Test 1, Test 2 and Test 3. Save this single screenshot as:
Test1_2_3_<your name>_<index number>

Take a screenshot of Test 4 and Test 5. Save this single screenshot as:
Test4_5_<your name>_<index number>

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

[5]

14. Save your program as **DICEROLLER2**_<your name>_<index number>**.py**

Extend your program to:

- When rolling the 20-sided die against the *Tn*,
    - a die roll of 1 is always a "`Fail`" even if the total value is greater than or equal to *Tn*.
    - a die roll of 20 is always a "`Pass`" even if the total value is less than *Tn*.
- Instead of ending, output "`Enter "NO" to quit, press enter to continue …`"
- Allow the user to input a string that will be converted to upper case by the program after entry.
- If the input string is "`NO`" then end the program; otherwise run the program again.

[3]

--- END OF PAPER ---