

BOON LAY SECONDARY SCHOOL

PRELIMINARY EXAMINATION

2018

Name	()
Class		

Subject	:	COMPUTING
Paper No	:	2
Subject Code	:	7155/02
Level	:	SECONDARY FOUR EXPRESS
Date/Day	:	28 AUGUST 2018 (TUESDAY)
Time	:	1130 - 1400
Duration	:	2 HOUR 30 MINUTES

Additional Materials: Electronic version of CARLOAN.XLSX data file Electronic version of TEXTCHECK.PY file Electronic version of JOB_ERROR.PY file Insert Quick Reference Glossary

READ THESE INSTRUCTIONS FIRST

Before you start your exam, check that you have received the correct paper and the number of printed pages are correct.

Write your name, index number and class in the spaces 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 number of marks for this paper is 50.

Task 1

Century Boon Lay Loan Service provides car loan to new car owners. The manager wants to use a spreadsheet software to record the details of the transactions in the first quarter of 2018. You are required to finish setting up the spreadsheet to record the interest earned.

Open the file **CARLOAN**. You will see the following data.

Save the file as **LOAN**_<your name>_<centre number>_<index number>.

	А	В	С	D	E	F		
1	Century Boon Lay Loan Service							
2								
	Car Loan		Years of	Interest		Free 1 st Year		
3	Transaction	Loan Amount	Loan	Rate	Interest Payable	Car Insurance		
4	CBL0115	\$65,000.00	7					
5	CBL0116	\$38,000.00	3					
6	CBL0117	\$123,000.00	10					
7	CBL0118	\$25,600.00	3					
8	CBL0119	\$32,100.00	5					
9	CBL0120	\$50,000.00	5					
10	CBL0121	\$43,800.00	5					
11	CBL0122	\$76,000.00	8					
12	CBL0123	\$38,000.00	5					
13	CBL0124	\$98,000.00	8					
14	CBL0125	\$21,300.00	3					
15	CBL0126	\$57,800.00	6					
16	CBL0127	\$104,000.00	10					
17	CBL0128	\$88,000.00	8					
18	CBL0129	\$65,700.00	5					
19								
20	Total							
	Number of Car							
21	Loan Transactions							
22	Range of years							
23					Rates			
				Years of	Description	Interest Rate		
24				Loan		per Year		
25				1	One to three	3.0%		
26				4	Four to six years	2.0%		
27				7	Seven to nine	1.8%		
28				10	Ten years or more	1.5%		
20								

1 In cell **B20** enter a formula to calculate the total loan amount.

[1]

[1]

- 2 In cell **B21** enter a formula to count the number of transactions made.
- 3 In cell **B22** enter a formula to calculate the range of the number of years loaned. [1]
- 4 Use an appropriate function to search for the **Interest Rate per Year** in the **Rates** table and use it to complete the **Interest Rate** column. [2]
- 5 Enter a formula to calculate the simple interest payable by car owners and use it to complete the **Interest Payable** column. Round these values to the nearest whole number. [3]

6 Enter a formula, by using conditional statement(s), to identify car owners who have loaned \$50,000 or more for over 5 years and put YES in the Free 1st Year Car Insurance column. Otherwise put NO in the Free 1st Year Car Insurance column. [2]

Save and close your file.

Task 2

The following program accepts five inputs and prints out the length of the longest input and the average length of all the inputs. An input may only contain letters, digits or symbols.

```
longest = 0
total = 0
data = 5
for count in range(data):
    text = input("Please enter the text: ")
    if len(text) > longest:
        longest = len(text)
    total += len(text)
print("The longest input length is ", longest)
average_length = total/data
print("Average length is ", average length)
```

Open the file TEXTCHECK.py

Save the file as **TENTEXT_**<your name>_<centre number>_<index_number>

7 Edit the program so that it:

(a)	Accepts 10 inputs	[1]			
(b)	Determines and prints out the length of the shortest input as well	[4]			
(c)	Tests if the input included empty space, if yes, asks the user for input again. An input can only contain letters, digits or symbols.	[3]			
Sav	/e your program.				
Save your program as VARTEXT <your name=""> <centre number=""> <index number=""></index></centre></your>					

[2]

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

Save your program.

8

Task 3

The following program should check which student is eligible for the job attachment programme using the following rules:

- age must be at least fourteen years
- age not more than eighteen years
- mid-year examination overall results greater than 60.0%.

The program calculates the number of students who are eligible for the job attachment programme and the number rejected. When a student is rejected, all the reason(s) for rejection will be printed. The program finishes when an age of zero or a mark of zero is input. The number of students who are eligible for the job attachment and the number rejected are then printed out.

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

```
age = 0
results = float(0)
rejected = 100
eligible = 0
age = int(input"Please enter student's age:" ))
result = float(input("Please enter mid-year exam overall results:"))
while age <> 0 and result <> 0:
    if age < 15 or age > 18 or result == 60:
        if age < 15:
            print("Age must be at least fourteen years")
        elif age > 18:
            print("Age must not be more than eighteen")
        if result == 60:
            print("Mid-year exam overall results must be more than 60%")
        rejected = rejected - 1
        print("Student is NOT eligible for job attachment programme.")
else:
        print("Student is eligible for job attachment programme.")
        Eligible = eligible + 1
        age = int(input("Please enter student's age:" ))
    result = float(input("Please enter mid-year exam overall results:"))
print("Number of students eligible is " eligible)
print("Number of students rejected is ", rejected)
```

Open the file JOB_ERROR.py

Save the file as JOB_<your name>_<centre number>_<index_number>

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

```
[10]
```

Save your program.

Task 4

You have been tasked to write a program to calculate the average number of parcels collected per week by a team of four postmen. Each postman can carry at most 10 parcels and there are 5 working days per week.

The program should allow you to:

- Enter data in the format of **a**, **b**, **c**, **d** where **a**, **b**, **c**, **d** are the number of parcels collected by each postman for the day. An example is 5, 8, 10, 9
- Only allow data entry of 0 to 10.
 Text message "Error Please enter number 0 to 10" should be displayed if any invalid data is entered
- Calculate the total number of parcels collected each day
- Repeat this for a total of the five working days per week
- Find the average number of parcels collected by a postman per week rounded to the nearest whole number
- Calculate the total number of parcels collected for the week
- Display the output data on the screen. Your output **must** look like this:

Enter parcels collected by the team: 2, 11, 5, 8

Error – Please enter number 0 to 10.

Enter parcels collected by the team: 5, 8, 10, 9

Enter parcels collected by the team: 0, 3, 8, 7

Enter parcels collected by the team: 1, 3, 5, 9

Enter parcels collected by the team: 6, 10, 2, 4

Enter parcels collected by the team: 2, 1, 8, 9

- Day 1 The team collected 32 parcel(s)
- Day 2 The team collected 18 parcel(s)
- Day 3 The team collected 18 parcel(s)
- Day 4 The team collected 22 parcel(s)
- Day 5 The team collected 20 parcel(s)

Average number of parcels				
Total number of parcels for the week	110			

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

Save your program as **POST1_**<your name>_<centre number>_<index_number>

- 11 When your program is working, use the following test data to show your test results:
 - 5, 8, 10, 9
 - 0, 3, 8, 7
 - 1, 3, 5, 9
 - 6, 10, 2, 4
 - 2, 1, 8, 9

Take a screen shot of your results and save it as a bitmap file**POSTRESULT**_<your name>_<centre number>_<index_number>[5]

12 Save your program as **POST2**_<your name>_<centre number>_<index_number> Extend your program to identify which postman collected less than 30 parcels in the week.

The additional output printed on the screen should look like this:

Postman 1 collected 14 parcels this week.[3]Postman 2 collected 25 parcels this week.[3]

Save your program.

 13 Save your program as POST3_<your name>_<centre number>_<index_number>

 Extend your program to work with any number of postmen.
 [2]

Save your program.

End of Paper