JUNYUAN SECONDARY SCHOOL
PRELIMINARY EXAMINATION 2019
SECONDARY FOUR EXPRESS

CANDIDATE NAME	

CLASS				INDEX NUMBER		

COMPUTING 7155/02

Paper 2 (Practical)

23 Aug 2019

2 hours 30 minutes

Additional Materials: 1 x Thumb Drive
Electronic version of TASK1.xls file
Electronic version of TASK2.py file
Electronic version of TASK3.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		
Task 1		10
Task 2		10
Task 3		10
Task 4		20
Total		50

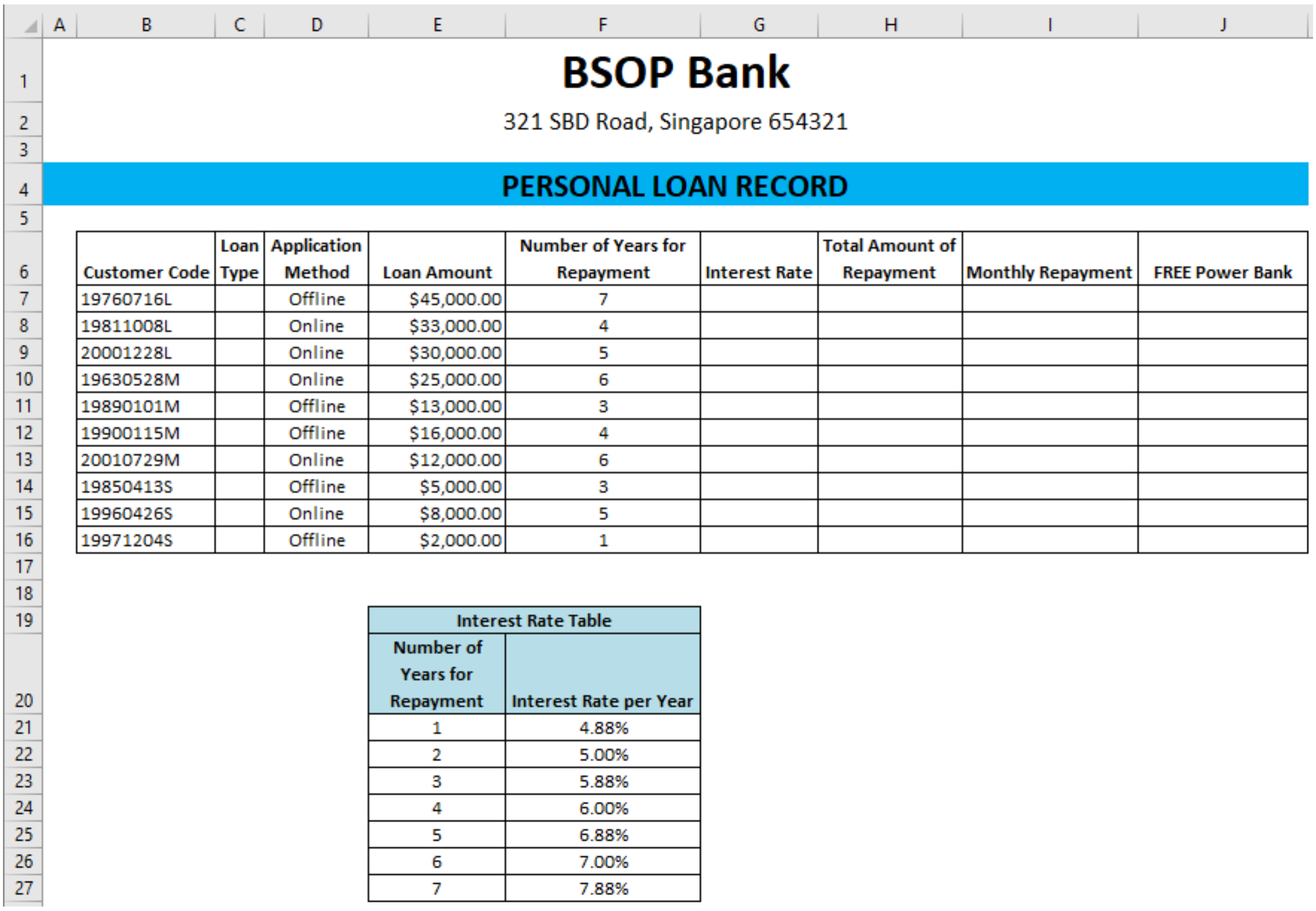
This document consists of 6 printed pages and 2 pages of Quick Reference for Python.

[Turn over

Task 1

BSOP Bank uses a spreadsheet for its customer loan records. You are required to finish setting up the spreadsheet to record the monthly repayment data.

Open the file **TASK1.xls.** You will see the following data.
Save the file as **LOAN_**<Class>_<Index>_<Name>**.xlsx**



BSOP Bank

321 SBD Road, Singapore 654321

PERSONAL LOAN RECORD

Customer Code	Loan Type	Application Method	Loan Amount	Number of Years for Repayment	Interest Rate	Total Amount of Repayment	Monthly Repayment	FREE Power Bank
19760716L		Offline	$45,000.00	7				
19811008L		Online	$33,000.00	4				
20001228L		Online	$30,000.00	5				
19630528M		Online	$25,000.00	6				
19890101M		Offline	$13,000.00	3				
19900115M		Offline	$16,000.00	4				
20010729M		Online	$12,000.00	6				
19850413S		Offline	$5,000.00	3				
19960426S		Online	$8,000.00	5				
19971204S		Offline	$2,000.00	1				

Interest Rate Table

Number of Years for Repayment	Interest Rate per Year
1	4.88%
2	5.00%
3	5.88%
4	6.00%
5	6.88%
6	7.00%
7	7.88%

1 For the cell range **C7:C16**, use an appropriate function to extract the last letter of the **Customer Code** to represent the **Loan Type**. [1]

2 Use an appropriate function to search for the **Interest Rate per Year** in the **Interest Rate Table** given and use it to complete the **Interest Rate** column. [2]

3 Enter a formula to calculate the simple interest payable by the customers and use it, together with the **Loan Amount**, to complete the **Total Amount of Repayment** column. [3]

4 Enter a formula to calculate the monthly amount repayable per customer and use it to complete the **Monthly Repayment** column. [2]

5 Use a conditional statement, to identify those customers who have undertaken a loan repayment amount of more than $15000 and whose application method was online, and put **YES** in the **Free Power Bank** column. Otherwise, put **NO** in the **Free Power Bank** column. [2]

Save and close your file.

Task 2

The following program accepts ten positive integers between 0 and 100, both exclusive, and prints out the smallest integer.

```
smallest = 100
counter = 0
while counter < 10:
    number = int(input("Enter a positive integer: "))
    if number < smallest:
        smallest = number
    counter += 1
print("The smallest integer is ", smallest)
```

Open the file **TASK2.py**

Save the file as **POSITIVE_**<Class>_<Index>_<Name>**.py**

6 Edit the program so that it:

(a) Accepts only five positive integers. [1]

(b) Prints out the largest integer as well as the smallest integer. [4]

(c) Tests if the number is positive, and if not, asks the user for another input. [3]

7 Save your program as **VARPOSITIVE_**<Class>_<Index>_<Name>**.py**

Edit your program so that it works for any number of inputs. [2]

Save your program.

Task 3

The following program should check whether a student is eligible for a work attachment programme using the following rules:

- age of student must be at least 15 years old,
- age of student must not be more than 17 years old,
- overall examination results greater than 70%.

The program calculates the number of students who are eligible for the work 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 work attachment and the number rejected are then printed out.
There are several syntax and logic errors in the program.

```
age, results, eligible = 0, 0, 0
rejected = 100
age = int(input"Please enter student's age:" ))
result = float(input("Please enter overall exam results:"))
while age <> 0 and result <> 0:
    if age < 16  or age > 17 or result == 70:
        if age < 15:
            print("Age must be at least 15 years old ")
        elif age > 17:
            print("Age must not be more than 17 years old ")
        elif result < 70:
            print("Overall exam results must be more than 70%")
        rejected = rejected - 1
        print("Student is NOT eligible for attachment programme.")
    else:
        print("Student is eligible for attachment programme.")
        Eligible = eligible + 1

    age = int(input("Please enter student's age:" )
    result = float(input("Please enter overall exam results:"))

print("Number of students eligible is " eligible)
print("Number of students rejected is ", rejected)
```

Open the file **TASK3.py**
Save the file as **WORK_**<Class>_<Index>_<Name>**.py**

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

Task 4

You have been asked to write a program for a department store to calculate the average and total number of jeans sold per week by a particular brand. The jeans come in four sizes, S, M, L and XL. The store only keeps 10 pieces of each size in stock for each day.

The program should allow you to:

- Enter data in the format a, b, c, d where a, b, c, d are the sizes, S, M, L, XL respectively that are sold at a store for a day (An example is 1, 2, 9, 10)
- Only allow data entry of 0 to 10 for each size sold each day
- Calculate the total number of jeans sold each day
- Repeat this for a total of seven days
- Find the average number of jeans sold per day rounded to the nearest whole number
- Calculate the total number of jeans sold for the week.
- Display this on the screen. Your output **must** look like this:

```
Day 1  Total jeans: 22
Day 2  Total jeans: 10
Day 3  Total jeans: 15
Day 4  Total jeans: 13
Day 5  Total jeans: 6
Day 6  Total jeans: 13
Day 7  Total jeans: 20
Average number of jeans sold per day: 14
Total number of jeans sold for the week: 99
```

9 Write your program and test that it works.
Save your program as **JEANS1_**<Class>_<Index>_<Name>. [10]

10 When your program is working, use the following test data to show your test results:

```
1, 2, 9, 10
2, 3, 4, 1
0, 10, 2, 3
5, 2, 0, 6
0, 1, 0, 5
3, 2, 1, 7
3, 7, 6, 4
```

Take a screen shot of your results and save the screenshot as:
JEANSRESULTS_<Class>_<Index>_<Name>.

Save your files in either **.png** or **.jpg** format. [5]

11 Save your program as **JEANS2_**<Class>_<Index>_<Name>.

Extend your program to identify the days when the sales of the jeans was poor. To be considered poor, the total sales for the day must be less than 15 jeans sold. Your output should look like this:

```
Poor Day 2  Total jeans: 10
Poor Day 4  Total jeans: 13
Poor Day 5  Total jeans: 6
Poor Day 6  Total jeans: 13
```

Save your program. [3]

12 Save your original answer in **Q9** as **JEANS3_**<Class>_<Index>_<Name>.

Extend your program to work for any number of days by asking the user to input the duration required.

Save your program. [2]

End of Paper