

### JUNYUAN SECONDARY SCHOOL PRELIM EXAMINATION 2018 SECONDARY FOUR EXPRESS

CANDIDATE NAME

CLASS

INDEX NUMBER



# COMPUTING

Paper 2 (Practical)

7155/02

26 July 2018 / Thursday 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

# **READ THESE INSTRUCTIONS FIRST**

Answer <u>ALL</u> questions.

All tasks must be done in the exam venue. 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					

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

	1.1					J			
- 4	A B	С	D	E	F	G	H	I	J
1					Maynot	Bank			
2	123 Maybe Road, Singapore 123456								
3					, , ,	0.			
4									
5									
6	PERSONAL LOAN RECORD								
7									
8									
		Loan	Application		Number of Years for		Total Amount of		
9	Customer Code	Туре	Method	Loan Amount	Repayment	Interest Rate	Repayment	Monthly Repayment	FREE \$50 Cash-back
10	L19760716		Offline	\$50,000.00	7				
11	L19811008		Online	\$35,000.00	6				
12	L20001228		Online	\$30,000.00	5				
13	M19630528		Online	\$20,000.00	5				
14	M19890101		Offline	\$11,000.00	4				
15	M19900115		Offline	\$17,500.00	3				
16	M20010729		Online	\$14,000.00	4				
17	S19850413		Offline	\$6,500.00	2				
18	S19960426		Online	\$9,500.00	3				
19	S19971204		Offline	\$3,000.00	1				
20									
21						-			
22				Interest Rate Table		_			
				Number of					
				Years for					
23				Repayment	Interest Rate per Year				
24				1	6.00%				
25				2	6.38%	_			
26				3	6.88%	1			
27				4	6.88%	-			
28				5	6.88%				
29				6	7.00%				
30				7	7.20%	1			

Open the file **TASK1.xls.** You will see the following data.

Save the file as LOAN\_<Class>\_<Index>\_<Name>.

1 For the cell range C10:C19, use an appropriate function to extract the first letter of the Customer Code to represent the Loan Type.

[1]

[2]

[3]

[2]

[2]

- 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.
- 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.
- 4 Enter a formula to calculate the monthly amount repayable per customer and use it to complete the **Monthly Repayment** column.
- Use a conditional statement, to identify those customers who have undertaken a loan repayment period of more than 3 years and whose application method was online, and put YES in the Free \$50 Cash-back column. Otherwise, put NO in the Free \$50 Cash-back column. Save and close your file.

The following program accepts the weight in kilograms (kg) for 10 students and prints out the largest weight and the average weight. The weights are in the range of 40 kg to 100 kg.

```
largest = 0
total = 0
student = 10

for i in range(student):
    weight = float(input("Enter weight of student: "))
    if weight > largest:
        largest = weight
    total = total + weight

average = round(total/student, 1)
print("Largest weight is ", largest, "kg")
print("Average weight is ", average, "kg")
```

Save your program as WEIGHT2\_<Class>\_<Index>\_<Name>.

#### Open the file TASK2.py

6	Edit the program, so that it: (a) Accepts the weight for 40 students.				
	(b) Prints out the smallest weight.	[4]			
	(c) Tests if the weight input is between 40 and 100 (both inclusive), and if not, inform the user that it is an invalid input and asks the user for input again as necessary.	[3]			
	Save your program as <b>WEIGHT1</b> _ <class>_<index>_<name>.</name></index></class>				
7	Edit your program so that it works for any number of students.	[2]			

3

The Singapore Flying Academy has a requirement that its student pilots must have a height between 165 cm and 190 cm (both inclusive). The following program prompts the user to enter the height of the applicant. The program will calculate and display the number of applicants that are accepted and the number of applicants that are rejected. The program uses the following rules:

- The height is entered as an integer value.
- The program will terminate when a height of 0 or no data is entered.

There are several syntax and logical errors in the program.

```
height=0
accepted=0
rejected=100
input_str=input("Enter height in cm: )
while input_str != "" or height != "0":
    height=int(input_str)
    if height > 165 and height < 190:
        accepted = accepted + 1
    else:
        rejected = rejected - 1
    input_string=input("Enter height in m: ")
print("Total accepted = ",height)
print("Total rejected = ",rejected)</pre>
```

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

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

Save your program.

You have been asked to write a program for a department store to calculate the average number of t-shirts sold per week by a particular brand. The t-shirts come in 5 sizes, XS, S, M, L and XL. The store only keeps 20 pieces of each size in stock for each day.

The program should allow you to:

- Enter data in the format a, b, c, d, e where a, b, c, d, e are the sizes, XS, S, M, L, XL respectively that are sold at a store for a day. An example is 10, 20, 9, 10, 0.
- Only allow data entry of 0 to 20 for each size sold each day, otherwise exit the program
- Calculate the total number of t-shirts sold each day
- Repeat this for a total of seven days
- Find the average number of shirts sold per day rounded to the nearest whole number
- Calculate the total number of shirts sold for the week.
- Display this on the screen. Your output **must** look like this:

```
Day 1 Total shirts: 49
Day 2 Total shirts: 47
Day 3 Total shirts: 26
Day 4 Total shirts: 60
Day 5 Total shirts: 25
Day 6 Total shirts: 63
Day 7 Total shirts: 29
Average number of shirts sold per day: 43
Total number of shirts sold for the week: 299
```

**9** Write your program and test that it works.

Save your program as **TSHIRT1**\_<Class>\_<Index>\_<Name>. [10]

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

```
10, 20, 9, 10, 0
2, 13, 4, 19, 9
0, 10, 2, 3, 11
15, 20, 0, 6, 19
0, 10, 0, 15, 0
13, 12, 11, 17, 10
3, 7, 6, 4, 9
```

Take a screen shot of your results and save it as bitmap **TSHIRTRESULTS\_**<Class>\_<Index>\_<Name>.

[5]

11 Save your program as **TSHIRT2\_**<Class>\_<Index>\_<Name>.

Extend your program to identify the sales of the t-shirts by their sizes. Print out the size of the t-shirt and the total number of that size sold in the week. Your output should look like this:

Size XS Total shirts: 43 Size S Total shirts: 92 Size M Total shirts: 32 Size L Total shirts: 74 Size XL Total shirts: 58

Save your program.

12 Save your program as TSHIRT3 <Class> <Index> <Name> The owner of the t-shirt brand decided to expand his t-shirt size range by changing the sizes to numeric values. E.g. 1, 2, 3, 4, 5, 6.

Extend your program to work for any numeric range of t-shirt sizes by asking the user to input the minimum and maximum sizes.

Save your program.

**End of Paper** 

[3]

[2]