Name_____ ( ) Class: Sec _____

SERANGOON SECONDARY SCHOOL

PRELIMINARY EXAMINATION 2018

SECONDARY 4 EXPRESS

COMPUTING **7155/02**

PAPER 2 **28 Aug 2018**

Name of Setter: **2 hour 30 minutes**

READ THESE INSTRUCTIONS FIRST

Write your name, class and index number on all the work you hand in.
Write in dark blue or black pen.
Answer **all** questions.

All tasks must be done in the computer laboratory.
Programs are to be written in Python. A Quick Reference Glossary for Python is provided.

The number of marks is given in brackets [ ] at the end of each question or part question.

Retrieve the template files from the **COMPEXAM** folder in the thumb drive.

Save your work inside the **COMPEXAM** folder in the thumb drive using the file name given in the question as and when necessary.

FOR EXAMINER'S
50

Quick Reference for Python

This quick reference shows some examples of the Python language constructs. The complete Python language is not limited to these examples.

1. Identifiers

When naming functions, variables and modules, the following rules must be observed:

- Names should begin with character 'a' - 'z' or 'A' - 'Z' or '_'
  and followed by alphanumeric characters or '_'
- Reserved words should not be used.
- User-defined identifiers are case sensitive.

2. Comments and Documentation Strings

```
# This is a comment

"""
   This is a documentation string
   over multiple lines
"""
```

3. Input/Output

```
print ("This is a string")

s = input ("Instructions to prompt for data entry.")
```

4. Import

import <module>

e.g. **import** math

5. Data Type

Data Type	Notes
int	integer
float	real number
Bool	boolean
str	string (immutable)
list	series of values

6. Assignment

Assignment Statement	Notes
a = 1	integer
b = c	variable
d = "This is a string"	string
mylist = [1, 2, 3, 5, 5]	list or array

7. Arithmetic Operators:

Operator	Notes
+ -	plus, subtract
* /	multiply, divide
%	remainder or modulus
**	exponential or power
//	quotient of floor division

8. Relational Operators:

Operator	Notes
==	equality
!=	not equal to
> >=	greater than, greater than or equal to
< <=	less than, less than or equal to

9. Boolean Expression

Boolean Expression	Notes
a and b	logical and
a or b	logical or
not a	logical not

10. Iteration

while loop
while condition(s): <statement(s)

for loop
for in range(n): <statement(s)>
for record in records: <statement(s)>

11. Selection

Type 1		Type 2		Type 3
if condition(s): <statement(s)>		**if** condition(s): <statement(s)> **else**: <statement(s)>		**if** condition(s): <statement(s)> **elif** condition(s): <statement(s)> **else**: <statement(s)>

12. Built-in functions

(a) Basic functions

abs( )	chr()	float()	input()	int()
ord( )	print( )	range( )	round( )	str( )
format( )				

(b) Mathematical functions

ceil( )	exp()	fabs()	floor()	log()
max( )	min( )	pow( )	sqrt( )	trunc( )
format( )				

(c) String functions

endswith( )	find()	isalnum()	isalpha()	digit()
islower( )	isspace( )	isupper( )	len( )	lower( )
startswith( )	upper( )			

13. Reserved Words

Reserved words cannot be used as identifiers. They are part of the syntax of the language.

False	None	True	and	as
assert	break	class	continue	def
del	elif	else	except	finally
for	from	global	if	import
in	is	lambda	nonlocal	not

Task 1

A sports event organiser uses a spreadsheet to record the height and weight of the participants. You are required to finish setting up the spreadsheet to record the health risk of the participants.

Open the file **BMI**. You will see the following data. Some of the rows in the file have been hidden in the display below so you can see the top and bottom.

Save the file as **BMI_PART**<class _name>_<index no>_<your name>

	A	B	C	D	E	F	G	H
1	**Participants BMI**							
2								
3	**Name**	**Age**	**Height (in metres)**	**Weight (in kg)**	**BMI**	**Risk**	**Healthy**	
4	FANG SHUHUI	43	1.86	56.8				
5	GLADYS YUEN JING WEN	28	2.14	53.1				
6	GOH XIN YI	24	2.07	80.2				
7	HAN JINGYU	37	2	73.2				
8	JANICE NEO WEI TING	50	1.44	92.7				
9	SNG JIA YUN	43	1.9	98.7				
10	THEA BHAVINA BABANI	22	1.26	78.1				
25								
26	Number of particpants							
27	Average age							
28	Age range							
29								
30						**BMI Range**		
31						**Bmi**	**Health Risks**	
32						0	Underweight	
33						18.5	Healthy weight	
34						23	Slightly Overweight	
35						27.5	Obese	
36								

1. In cell **B26** enter a formula to count how many participants are there in the event. [1]

2. In cell **B27** enter a formula to count calculate the average age of the participants. [1]

3. In cell **B28** enter a formula to calculate the age range of the participants. [1]

4. The formula to calculate BMI is given as $BMI = kg / m^2$ where *kg* is the weight (in kg) and m^2 is the square of the height in metres. Use the formula to complete the **BMI** column. Round these values to one decimal place. [3]

5. Use an appropriate function to search for the **Health Risk** in the **BMI Range** table and use it to complete the **Risk** column. [2]

6. Use a conditional statement, to identify those participants who are at risk. Put a **YES** in the **Healthy** column if the participant has a healthy weight. Otherwise put **NO** in the **Healthy** column and format the cells with a **NO** to automatically show a red background. [2]

Save and close your file.

Task 2

The following program accepts the age for 20 people and prints out the average age.

```
total=0
ppl=20
for i in range(ppl):
    age = int(input("Enter age : "))
    total = total + age
average = total/ppl
print("Average age is ", average)
```

Open the file **AGE.py**

Save the file as **PAGE_**<class _name>_<index no>_<your name>.

7. Edit the program so that it:

(a) Accepts age for 50 people. [1]

(b) Prints out the number of people whose age are below 18 years old. [4]

(c) Tests if the age input is between 15 to 65 (inclusive), and if not, asks the user for input again as necessary. [3]

Save your program.

8. Save your program as **VARAGE**_<class _name>_<index no>_<your name>.

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

Save your program.

Task 3

The following program should check who can become a GRAB driver using the following rules:

- age at least twenty-one years old
- minimum two years of driving experience
- hold a valid Private Hire Car Driver's Vocational License (PDVL)
- Singapore citizen

The program calculates the number of people who qualify to be a GRAB driver and the number rejected. The program finishes when an age of zero is input. The number of people who qualify to be GRAB driver and the number rejected are then printed out.

There are several syntax and logical error(s) in the program.

```
age = 0
exp = 0
pdvl = "N"
citizen = "N"
rejected = 0

age = int(input("Please enter your age " ))
exp = int(input("Please enter years of driving experience " ))
pdvl = input("Do you hold a Private Hire Car Driver Vocational License
           (PDVL): (Y/N) ")
citizen = int(input("Are you a Singapore Citizen (Y/N) "))
while age == 0:
    if age < 21 or exp > 2 or pdvl == "N" or citizen == "N":
        if age <= 21:
            print("You are too young to drive")
        if exp < 2:
            print("You do not have enough driving experience")
        if pdvl == "N":
            print("You need to obtain a PDVL license first")
        else citizen == "N":
            print("You have to be a Singapore Citizen")
        rejected -= 1
    elif:
        print("You qualify to be a GRAB driver ")
        qualify = qualify + 1
    age = int(input("Please enter your age " ))
    exp = int(input("Please enter years of driving experience "))
    pdvl = input("Do you hold a Private Hire Car Driver Vocational License
                (PDVL): (Y/N) ")
print("Number of people rejected ", qualify)
print("Number of people qualified ", qualify)
```

Open the file **GRAB.py**

Save the file as **GRAB**_<class _name>_<index no>_<your name>

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

Task 4

You have been asked to write a program to calculate the average three-day temperature in Singapore. The daily temperature reading is taken at four different locations in Singapore.

The program should allow you to:

- Enter data in the format *n, s, e, w* where *n, s, e, w* indicates the temperature reading at four locations in Singapore. An example 29.7, 30.5, 31.6, 29.5
- Only allow data entry of one decimal place between 0.0 to 50.0 degrees celcius
- Calculate the average daily temperature rounded to one decimal place
- Repeat this for a total of three days
- Find the average three-day temperature rounded to one decimal place.
- Display this on the screen. Your output **mus**t look like this:

```
Day 1    Average Temperature 30.3
Day 2    Average Temperature 29.8
Day 3    Average Temperature 30.3

Average 3-day temperature:  30.1
```

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

Save your program as **TEMP1**_<class _name>_<index no>_<your name>

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

29.7, 30.5, 31.6, 29.5
28.8, 29.9, 30.1, 30.3
30.2, 30.0, 30.2, 30.9

Take a screen shot of your results and save it as a bitmap
TEMPRESULTS_<class _name>_<index no>_<your name>. [5]

12. Save your program as **TEMP2**_<class _name>_<index no>_<your name>.
Extend your program to identify each location where the temperature readings were more than 30.0 degrees on two or more occasions within the three-day period. Print out the location and the number of occasions where the temperature was more than 30.0 degrees celcius. Your output should look like this:

```
Location 3    3 occasions
Location 4    2 occasions
```
[3]

Save your program.

13. Save your program as **TEMP3**_<class _name>_<index no>_<your name>.
Extend your program to work with any number of locations. Save your program. [2]

END OF PAPER