ST. PATRICK'S SCHOOL
PRELIMINARY EXAMINATIONS 2018
SECONDARY 4 EXPRESS

NAME

CLASS | INDEX NUMBER

COMPUTING 7155/02

Paper 2 Practical (Lab-based) **13 September 2018**

2 h 30 min

Additional Materials:
- Electronic version of JOBS.XLSX data file
- Electronic version of ANALYSIS.PY file
- Electronic version of ROBOT.PY file
- Insert Quick Reference Glossary

READ THESE INSTRUCTIONS FIRST

Write your name, class, index number 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.

Parent's Signature : _____

Date: _____

Remarks (if any) :

For Examiner's Use	
Marks	/50
Total	**%**

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 the floor division

8. Relational Operators:

Operator	Notes
==	equality
!=	real number
> >=	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 i **in** range(n): <statement(s)>
for record **in** records: <statement(s)>

11. Selection

Type 1

```
if condition(s):
    <statement(s)>
```

Type 2

```
if condition(s):
    <statement(s)>
else:
    <statement(s)>
```

Type 3

```
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()

(c) String functions

endswith( )	find()	isalnum()	isalpha()	isdigit()
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
or	pass	raise	return	try
while	with	yield		

Task 1

A solutions company uses spreadsheet software to record its jobs details. You are required to finish setting up the spreadsheet.

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

Maintenance Pte Ltd

Job Details

No.	Job type	Expertise Required	Job ID	Job Code	Hourly Rate	Hours Required	Cost
1	Plumbing	Home Maintenance	ABC001234-HM		15	2	
2	Electrical	Electrical Engineering	ABC002234-EE		12	2	
3	Carpentry	Home Maintenance	ABC001235-HM		18	4	
4	Engineering Design	Mechanical Engineering	ABC002235-ME		22	4	
5	Business Reports	Business Administration	ABC003234-BA		10	5	
6	Interior Renewal	Home Maintenance	ABC001236-HM		20	3	
7	Marketing	Business Administration	ABC003234-BA		8	4	
8	Stock Inventory	Retail Sales	ABC004234-RS		7	3	
9	House Cleaning	Home Maintenance	ABC001237-HM		13	6	
10	Laboratory Cleaning	Chemical Processing	ABC005234-CP		17	3	

Job Code Fees

Job Code	Fees
HM	10
EE	25
BA	20
CP	30
RS	22
ME	27

Staff Required

Expertise	No.of Staff
Home Maintenance	
Electrical Engineering	
Mechanical Engineering	
Business Administration	
Retail Sales	
Chemical Processing	

1 Save the file as **MYJOBS**_<Class>_<Class_Index_Number>_<Your_Name>

2 In the **Job Code** column, enter a function to extract the last 2 letters of the corresponding **Job ID.** [2]

3 In the **Cost** column, enter a function to look up the **Fees** in the **Job Code Fees** table, add that with the corresponding **Hourly Rate**, and multiply it by **Hours Required**, i.e. [Cost = Job Code Fees + (Hourly Rate x Hours)] [3]

4 Format the **Cost** column to automatically highlight those cells whose total cost is above $50 with a blue background. [1]

5 In the **Staff Required** table, enter a function in the **No. of Staff** column to compute the number of staff required for each expertise. [2]

6 Enter "Total Cost" in **G14** and bold it. In **H14**, enter a function to calculate the total cost of all the jobs. [2]

Save and close your file.

Task 2

The following program analyses a user password consisting of only digits and letters. The program then displays the list of digits and letters in the password.

```
digits = []
letters = []
password = input("Enter the password: ")

for index in range(len(password)):
    if password[index].isdigit():
        digits.append(int(password[index]))
    elif password[index].isalpha():
        letters.append(password[index])

print("Digits in the password: ", digits)
print("Letters in the password: ", letters)
```

7 Open the file **ANALYSIS.py**

Save the file as **MYANALYSIS**_<Class>_<Class_Index_Number>_<Your_Name>

Edit the program so that it:

(a) Displays the largest digit in the password. [1]

(b) Displays the list of uppercase letters. [2]

(c) Tests if the user password input is exactly 6 characters in length. If not, the program asks the user for input again with a corresponding error message. [3]

Save your program.

8 Save your program as **VARANALYSIS**_<Class>_<Class_Index_Number>_<Your_Name>

Edit your program so that it displays the average of all digits in the password. [4]

Save your program.

Task 3

The following program should display the current location of a robot and the distance travelled, after a series of movement instructions by the user.

The program asks the user to enter the direction and the number of steps for each move. The program ends when the user types “exit”. The program then displays the current location of the robot and the distance travelled.

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

```
number_steps
current_location = [0,0]
print("Current location: ", current_location)

while True:
    direction = input("Enter the direction (up, down, left, right).
Type exit to end: ")
    if direction == exit:
        continue
    else:
        number_steps = input("Enter the number of steps to move: ")
        if direction = "up":
            current_location[1] += number_steps
        elif direction == "down":
            current_location[1] -= number_steps
        elif direction == "left":
            current_location[0] += number_steps
        else if direction == "right":
            current_location[1] += number_steps
        print("\nCurrent location: ", current_location)
        print("Distance travelled: ",
round(math.sqrt(current_location[0]**2 + current_location[1]**2),2)
```

Open the file **ROBOT.py**

9 Save the file as **MYROBOT**_<Class>_<Class_Index_Number>_<Your_Name>

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 asked to write a program to record and analyse the competitive timing of swimmers in the men's heats of a National Swimming Championship.

The program should allow you to:

- Enter data in the format a, b, c, d where a, b, c, d are the time taken in seconds by a swimmer to complete a 100-metre race in the butterfly, backstroke, breaststroke and freestyle categories respectively. An example is `54.42, 56.00, 64.25, 51.01`
- Only allow data entry of timing between `40.00` seconds to `70.00` seconds inclusive for any of the 100-metre categories.
- Calculate the projected 400-metre individual medley timing of the swimmer by taking the sum of the swimmer's race timing across all four 100-metre categories.
- Repeat this for a total of seven swimmers.
- Find the average swimmer race timing, rounded off to two decimal places, for each of the 100-metre categories.
- Determine the champion for each of 100-metre categories.
- Display this on the screen. Your output **must** look like this:

```
Category       Ave Timing   Champion    Champ Timing
Butterfly      57.32        Swimmer 5   53.53
Backstroke     55.47        Swimmer 2   53.20
Breaststroke   63.00        Swimmer 5   59.88
Freestyle      54.28        Swimmer 1   51.01

Projected 400-metre Individual Medley Timing
Swimmer 1   225.68
Swimmer 2   226.91
Swimmer 3   234.54
Swimmer 4   240.12
Swimmer 5   219.04
Swimmer 6   221.83
Swimmer 7   242.42
```

10 Write your program and test that it works.

Save your program as **HEATS1**_<Class>_<Index_Number>_<Your_Name> [10]

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

```
54.42, 56.00, 64.25, 51.01
58.02, 53.20, 61.33, 54.36
60.12, 54.23, 62.16, 58.03
59.20, 58.34, 66.20, 56.38
53.53, 54.01, 59.88, 51.62
55.46, 53.43, 60.77, 52.17
60.51, 59.11, 66.39, 56.41
```

Take a screen shot of your results and save it as a bitmap
HEATSRESULTS_<Class>_<Index_Number>_<Your_Name> [3]

12 Save your program as **HEATS2**_<Class>_<Index_Number>_<Your_Name>
Extend your program to allow race timing entered to be "-" if a swimmer happens not to be participating one or more of the 100-metre category, and if so, that particular swimmer would not have the projected 400-metre individual medley timing.

Use the following to show your test results:

```
54.42, 56.00, 64.25, 51.01
58.02, -, -, 54.36
-, 54.23, -, 58.03
59.20, -, -, 56.38
53.53, 54.01, 59.88, 51.62
55.46, 53.43, 60.77, 52.17
60.51, 59.11, 66.39, 56.41
```

Your output **must** look like this:

```
Category        Ave Timing    Champion     Champ Timing
Butterfly       56.86         Swimmer 5    53.53
Backstroke      55.36         Swimmer 6    53.43
Breaststroke    62.82         Swimmer 5    59.88
Freestyle       54.28         Swimmer 1    51.01

Projected 400-metre Individual Medley Timing
Swimmer 1    225.68
Swimmer 2    -
Swimmer 3    -
Swimmer 4    -
Swimmer 5    219.04
Swimmer 6    221.83
Swimmer 7    242.42
```

[5]

13 Save your program as **HEATS3**_<Class>_<Index_Number>_<Your_Name>
A swimming competition would only be held if there are at least four participants.
Extend your program to ensure that there are at least four participating swimmers.
Save your program. [2]

End of Paper