ST. PATRICK'S SCHOOL PRELIMINARY EXAMINATION 2021				
SUBJECT :	Computing Paper 2 (7155/02)	DATE :	24 August 2021	
LEVEL : ADDITIONAL MATERIAL S.	Secondary 4 Express Electronic version of ORDERS.x1		2 hours 30 minutes	

MATERIALS: Electronic version of DRINK.py file Electronic version of REVERSE.py file Quick Reference Glossary

Candidates answer on the Question Paper.

# **INSTRUCTIONS TO CANDIDATES:**

Write your Name, Class and Index No. in the spaces at the top of this page. Write in dark blue or back pen.

Answer **all** the 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.

# **INFORMATION FOR CANDIDATES:**

The allocated marks for each question are indicated in the brackets [ ].

For Examiner's Use		
Score	/50	

This question paper consists of 8 printed pages, including the cover page.

# ANNEX A

## **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, 4, 5]	list or array

#### 7. Arithmetic Operators

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

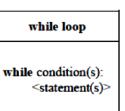
### 8. Relational Operators

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

#### 9. Boolean Expression

<b>Boolean Expression</b>	Notes
a and b	logical and
a or b	logical or
not a	logical not

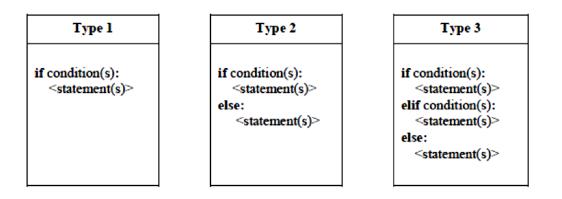
### 10. Iteration



for i in range(n): <statement(s)>

for record in records: <statement(s)>

### 11. Selection



### 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		

A food and beverage company uses a spreadsheet software to keep track of its customers' orders.

	А	В	С	D	E	F	
1	Tuesday, A	ugust 24, 2021					
2	Orders						
3	Order ID	Zone	Bento Set	Quantity	Total Price	Price After Discount	
4	E5969		С	3			
5	W2903		В	2			
6	N4090		А	1			
7	E5000		В	2			
8	S5904		D	3			
9	C4231		E	4			
10	W2144		А	2			
11	N5225		А	3			
12	C4119		E	2			
13	C1825		С	4			
14	S4616		В	2			
15	W2660		В	3			
16					Total Revenue		
17							
18	Bento Set	Α	В	С	D	E	
19	Price	\$20.40	\$22.60	\$32.30	\$17.90	\$23.50	
20	Total Quantity						

Open the file **ORDERS.xlsx**. You will see the following data.

Save the file as **MYORDERS\_**<Class>\_<Class\_Index\_Number>\_<Your\_Name>.xlsx

1	The first character of <b>Order ID</b> denotes the zone of the order. In cells <b>B4</b> to <b>B15</b> , enter a formula that uses an appropriate function to display the zone of the order.	[1]

In cells E4 to E15, enter a formula that uses an appropriate function to search for the Price of the respective Bento Set and calculate the total price of each order. [2]

3	A discount of 20% is given for orders whose <b>Total Price</b> is above \$40.	
	In cells F4 to F15, enter a formula that uses an appropriate function to calculate the price after	
	discount for each order.	[3]

- 4 In cell **F16**, enter a formula that uses an appropriate function that calculates the total revenue for the day. [1]
- 5 In cells **B20** to **F20**, enter a formula that uses an appropriate function that calculates the total quantity of each bento set ordered. [2]
- 6 In cells F4 to F15, use a formatting tool to change the colour of the cells whose value is at least \$50, to green fill.
   [1]

The following program displays a list of drinks available and asks user to input a drink to search within the list.

Open the file **DRINK.py** 

Save the file as MYDRINK\_<Class>\_<Class\_Index\_Number>\_<Your\_Name>.py

7 Edit the program so that it searches the list based on the user input and displays a message "Drink found. Enjoy!" when the drink is found within the list and "Drink not found." when the drink is not found.

### 8 Save your program as VARDRINK\_<Class>\_<Class\_Index\_Number>\_<Your\_Name>.py

Edit the program so that it:

- (a) Asks the user for input whether or not to remove the drink from the list after the drink is found within the list. If user inputs "Yes", then, the drink is removed from the list and the same updated list is displayed to the user.
- (b) Ask the user repeatedly to input a drink to search within the list (and also input whether or not to remove the drink from the list) until the letter "q" is entered or the list is empty. The program will then exit.

Save your program.

[4]

[3]

The following program reverses every word in a sentence.

For example, if the sentence "Thank you" is entered, the program outputs "knahT uoy".

There are several syntax and logic errors in the program.

```
input text = input("Enter input text: "
word list
current word = ""
result == ""
index = 0
while index > len(input text)
    if not input text[index].isspace():
        current word += input text[index]
    else:
    if current_word != "":
            word list += [current word]
            current word = ""
    index += 1
if current word != "":
    word list =+ [current word]
index = 0
while index < len(word list):</pre>
    current word = ""
    letter index = 0
    while letter index < len(word list[index]):</pre>
        current word += word list[index][-letter index - 1]
        letter index += 1
    word list[index] = current word
    index -= 1
index = 0
while index < (word_list):</pre>
    result += word list[index] + " "
    index += 1
result = result[:1]
output(result)
```

Open the file **REVERSE.py**.

Save the file as MYREVERSE \_<Class>\_<Class\_Index\_Number>\_<Your\_Name>.py

**9** Identify **and** correct the errors in the program so that it works according to the requirements given.

Save your program.

[10]

Moving averages are one of the most commonly used technical indicators by traders to identify price trends. They are normally used to filter out market noise and smooth out fluctuations in price to help traders distinguish between typical market fluctuations and actual price reversals.

The most basic type of moving average is Simple Moving Average (SMA). It is simply the average price over the specified period. The chart below shows an example of 200 period SMA, i.e. average over 200 data points.



Source: https://www.fidelity.com/learning-center/trading-investing/technicalanalysis/technical-indicator-guide/sma

### Example:

If the most recent 5 prices are \$1, \$2, \$3, \$4 and \$5, the average would be the sum of the rates i.e. (1+2+3+4+5) divided by the number of reporting periods. Therefore SMA = 15/5 =\$3

Each time a new price becomes available, the average "moves" so that the average is always based only on the last same number of variables. In this case, if the next number in the sequence is 6, the oldest rate, i.e. 1 would be dropped and the new average would (2+3+4+5+6)/5 which equals 4.

You are tasked to write a program that generates the SMA given user inputs of 5 data points for each of the 3 different commodities (Commodity A, Commodity B and Commodity C). The program should:

- For each commodity, take as input 5 data points in the format a, b, c, d, e where a, b, c, d and e are the prices across 5 reporting periods. You may assume that the prices are positive integers.
- Only allow data entry of positive integers.
- If user enters invalid data points, output an appropriate error message and ask the user to enter the data points again.
- Calculate the SMA using the formula given.
- Display the 5 data points and the SMA.
- Repeat the above steps for the 3 different commodities, i.e. Commodity A, Commodity B and Commodity C.
- 10 Write your program and test that it works. Save your program as **MYSMA**\_<Class>\_<Class\_Index\_Number>\_<Your\_Name>.py

[10]

- 11 When your program is complete, use the following test data to show your test result.
  - Commodity A with 1, 2, 3, 4, 5 as its data points across 5 reporting periods.
  - Commodity B with 3, 6, 4, -2, 2 as its data points across 5 reporting periods.
  - Commodity B with 3, 6, 4, 2, 2 as its data points across 5 reporting periods.
  - Commodity C with 4, 5, 9, 12, 20 as its data points across 5 reporting periods.

Take a screenshot of your result and save it as: **SMATEST\_**<Class>\_<Class\_Index\_Number>\_<Your\_Name>

Save your file in either .png or .jpg format.

12 Save your program as **MYSMA2\_**<Class>\_<Class\_Index\_Number>\_<Your\_Name>.py

Extend your program so that the average "moves" as described on the previous page. It should accept user input of the next data point in the format a, b, c where a is the next data point for Commodity A, b as the next data point for Commodity B and c as the next data point for Commodity C. The program should then calculate the new SMA for each of the 3 commodities.

The program should also display the initial data points and its initial SMA, as well as the new data points and its new SMA, for each of the 3 commodities.

Save your program.

[5]

[5]