

SERANGOON SECONDARY SCHOOL PRELIMINARY EXAMINATION SECONDARY 4 EXPRESS

CANDIDATE NAME				()	CLASS		
CENTRE NUMBER	S					INDEX NUMBER		

COMPUTING

Paper 2 (Lab-based)

7155/02 26 August 2020 2 hours 30 minutes

Additional Materials:	Electronic version of STBANK.xlsx data file
	Electronic version of USERNAME.py file
	Electronic version of FRUIT.py file
	Insert Quick Reference Glossary

Setter(s):

READ THESE INSTRUCTIONS FIRST

Rename the COMPEXAM folder in the thumb drive to <your name>_<centre number>_<index number>. Retrieve the electronic files from this folder in the thumb drive. Save your work inside this folder in the thumb drive using the file name given.

Answer all questions.

All tasks must be done in the computer laboratory. You are not allowed to bring in or take any pieces of work or materials on paper or media in any 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.



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	Notes
Statement	
a = 1	integer
$\mathbf{b} = \mathbf{c}$	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
< <=	less than, less than or equato

9. Boolean Expression

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

10. Iteration

while loop	for loop
while condition(s):	for in range(n):
<statement(s)< td=""><td><statement(s)></statement(s)></td></statement(s)<>	<statement(s)></statement(s)>
	for record in records:
	<statement(s)></statement(s)>

11. Selection

Type 1	Type 2	Туре 3
<pre>if condition(s): <statement(s)></statement(s)></pre>	<pre>if condition(s): <statement(s)> else: <statement(s)></statement(s)></statement(s)></pre>	<pre>if condition(s): <statement(s)> elif condition(s): <statement(s)> else: <statement(s)></statement(s)></statement(s)></statement(s)></pre>

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()	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 lam		nonlocal	not	

A bank uses a spreadsheet software to record the amount of loan for each client. You are required to finish setting up the spreadsheet to record the lump sum repayment amount.

	Α	В	C	D	E	F	G	Н	1	J	
	ST Bank Clients										
1											
					Number of		Compound	Lump Sum		Discounted	
2	Client ID	First Name	Last Name	Amount Loan	Vears	Client Type	Interest Rate	Renavment	Discount	Lumn Sum	
- 3	C19750346	Neil	McDonald	\$12 500.00	10	chefter type	interest nute	Repayment	Discount	Lump Sum	
5	C19812594	Christopher	Young	\$18,000,00	25						
6	\$19831434	lan	Vaughan	\$21,500,00	19						
7	\$19876607	Lauren	Murray	\$10,000,00	25						
8	C19767871	Amy	Dowd	\$25,000.00	23						
9	C19776832	Charles	Fisher	\$16,700.00	12						
10	C19905401	Jasmine	Harris	\$11.500.00	25						
11	S19895187	Julia	Graham	\$22,000.00	19						
12	S19891191	Vanessa	Bower	\$15,000.00	18						
13	S19948521	Bernadette	Nolan	\$17,200.00	11						
14	S19942376	John	Baker	\$13,500.00	20						
15	\$19835802	Vanessa	Lawrence	\$24,000.00	17						
16	S19728956	Luke	Hart	\$27,500.00	10						
17	C19867342	Sebastian	Pullman	\$10,900.00	14						
18	S19776434	Alison	Nolan	\$14,000.00	19						
19											
20	Median Amount Loaned				Number of Privileged clients						
21											
22											
23	Compound Rates per Ye			ates per Year							
			Number of	Interest Rate							
24			Years Loaned								
25	1		10	3.5%							
26	1		15	3.0%							
27	1		20	2.5%							
28	_		25	2.0%							
29	1										

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

Save the file as **STCLIENTS**_<your name>_<centre number>_<index number>.xlsx

- 1. In cell **D20** enter a formula to calculate the median amount of all the loans, rounded to [2] the nearest dollar (\$).
- In cells F4 to F18 enter a formula that uses an appropriate function to indicate the Client [1]
 Type (S for privileged clients and C for normal clients) by extracting the first letter from the Client ID column.
- 3. In cells G4 to G18 enter a formula that uses an appropriate function to search for the [2] Interest Rate in the Compound Interest Rate per Year table and use it to complete the Compound Interest Rate column.
- **4.** In cells **H4** to **H18** enter a formula that uses an appropriate function to calculate the [2] lump sum repayment amount for each client.

- 5. In cells **I4** to **I18** enter a formula to calculate the discount for each client. Each privileged [1] client is given a discount of **10%** of the lump sum repayment amount. Otherwise the discount is 0 (zero).
- 6. In cells J4 to J18 enter a formula to calculate the Discounted Lump Sum. It is the [1] Lump Sum Repayment less the Discount.
- 7. In cell **120** enter a formula to count the number of privileged clients. [1]

Save and close your file.

Task 2 begins on the next page.

The following program creates a username for a user. It creates the username by appending the name with a series of character '0' (zero) until the length of the username is ten characters long.

The name entered must be entered without any space, for example, JohnTan.

```
name = input("Please enter your name: ")
size = len(name)
if size < 10:
    username = name + (10-size)*'0'
print("Your username is " + username)</pre>
```

Open the file USERNAME.py

Save the file as MYUSERNAME_<your name>_<centre number>_<index number>

- **8.** Edit the program so that the username is created by appending the name with a series [1] of character '9' until the length of the username is ten characters long.
- **9.** The program needs to validate the length of the name and whether the name contains alphabets only.

Save the file as MYUSERNAME2_<your name>_<centre number>_<index number>

- (a) Edit the program to:
 - test whether the user has entered a name of between three to twenty characters containing only alphabets
 - output a suitable error message that asks the user to re-enter a name again, [4] and repeat this until the user enters a valid name.
- (b) Edit the program to:
 - ask the user to enter a single character ('0' to '9') to be used for appending the name with a series of this character until the length of the username is ten characters long
 - output a suitable error message that asks the user to enter a single digit again, and repeat this until the user enters a valid single digit
 - use the first ten characters of the name as the username if the name entered [5] contains ten or more characters.

Save your program.

The following program searches a list of fruits sorted in ascending order to check if it is sold in a shop.

If the fruit is found in the list, a message is displayed on the screen that states the number of items that are in the list and the position of the fruit in the list. Otherwise, a message is displayed on the screen that states the fruit is not in the list.

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

```
#list of fruits sorted in ascending order
fruits_list = ["apple", "grape", "guava", "mango", "orange", "pear"]
fruit_to_find = input("Which fruit would you like to search for? "
items = length(fruits_list)
found = True
for i in range(item):
    if fruits_list[index] == fruit_to_find:
        print("There are " + str(items) + " fruits in the list, " +
            fruit_to_find + " is item " + str(i) +
                " in the list.")
        found = True
    else fruits_list[i] < fruit_to_find:
        break
if found = True:
    print("The fruit is not in the list.")
```

Open the file FRUIT.py

Save the file as MYFRUIT_<your name>_<centre number>_<index number>

10. Identify and correct the errors in the program.

[10]

Save your program.

You have been asked to create a program to generate a check digit.

The program must:

- Allow a user to enter a twelve-digit number. The program must ask for another twelve-digit number each time the user enters an input that is not a twelve-digit number.
- Each digit, starting from left to right, is alternately multiplied by the weights 1 or 3 and the results are sum up together. Example of a twelve-digit : 978030640615 9×1 + 7×3 + 8×1 + 0×3 + 3×1 + 0×3 + 6×1 + 4×3 + 0×1 + 6×3 + 1×1 + 5×3 = 93
- The sum from the previous step is divided by ten to get the remainder. 93 / 10 = 9 remainder 3
- Subtract the remainder from ten to obtain the check digit. 10 - 3 = 7So 7 is the check digit.
- Display this on the screen. Your output must look like this:

7 is the check digit for the twelve-digit number 978030640615

11. Write your program and test that it works.

Save your program as CD1 <your name> <centre number> <index number>.py

- 12. When your program is working, test it for the following :
 - Test 1 User enters the number 97803064061
 - Test 2 User enters the number 9780306406157
 - Test 3 User enters the number 97803064A615
 - Test 4 User enters the number 979861499524
 - Test 5 User enters the number 057294690123

Take a screenshot of Test 1, 2 and 3. Save this **single** screenshot as: TEST123 <your name> <centre number> <index number>

Take a screenshot of Test 4 and 5. Save this **single** screenshot as: TEST45 <your name> <centre number> <index number>

Save you files in either .png or .jpg format.

13. Save your program as **CD2** <your name> <centre number> <index number>.py

Extend your program to allow the user to input the weights for both the odd-numbered and even-numbered digit to be used for the calculation of the check digit.

You are not required to validate the weights.

[4]

Save your program.

14. Save your program as CD3_<your name>_<centre number>_<index number>.py

Extend your program to allow the user to enter either a twelve-digit or thirteen-digit number. When the user enters a thirteen-digit number, the right most digit is the check digit and a suitable message **must** then be output to indicate whether this check digit is valid or not.

You are not required to do any additional validation.

Save your program.

END OF PAPER

[4]