**BOON LAY SECONDARY SCHOOL** 



# PRELIMINARY EXAMINATION

2021

Name	
CCA	

Subject	:	COMPUTING
Paper No	:	2 (LAB-BASED)
Subject Code	:	7155/02
Level	:	SECONDARY FOUR EXPRESS
Date/Day	:	15 <sup>th</sup> SEPTEMBER 2021 / WEDNESDAY
Time	:	0815 – 1045
Duration	:	2 HOURS 30 MINUTES

Additional Materials: Electronic version of Total Residents (2010-2019).xlsx Electronic version of HANGMAN.py Electronic version of NUMBER.py Electronic version of Insert Quick Reference Glossary

## **READ THESE INSTRUCTIONS FIRST**

Before you start your exam, check that you have received the correct paper and the number of printed pages are correct.

Write your name, index number, and CCA in the spaces at the top of this page.

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.

This document consists of **6** printed pages.

### Task 1

The figure below shows the total residents in Singapore between 2010 and 2019.

	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
0 - 4 Years	194432	188249	186673	183295	181369	183575	187160	187653	185528	185355
5 - 9 Years	215675	208119	206324	205704	205790	204452	201509	200575	199066	197775
10 - 14 Years	244302	240536	233320	226244	220796	214388	207495	206253	206530	207926
15 - 19 Years	263750	260738	258861	255066	247337	242902	239771	232973	226520	222222
20 - 24 Years	247190	255069	264506	266889	263918	264127	260854	259072	255516	249079
25 - 29 Years	272639	262606	254582	255615	265659	271030	279988	290198	291631	288194
30 - 34 Years	298687	297555	295396	296870	292287	290619	285544	279340	281397	293446
35 - 39 Years	320024	316161	313970	305503	302029	301067	301998	300956	303720	300818
40 - 44 Years	309441	306614	308823	312581	316023	316755	313445	311484	303640	300763
45 - 49 Years	323459	323991	320925	316336	308533	303413	301183	303457	307838	311682
50 - 54 Years	303044	308130	310376	312988	315503	315091	315598	312814	308965	301682
55 - 59 Years	248696	260678	271817	281069	288392	295063	299591	301678	304390	306759
60 - 64 Years	191995	208160	213996	222154	231502	240493	251853	262651	271688	278828
65 - 69 Years	111511	112396	128879	145817	161198	182425	198020	203969	212101	221296
0 - 19 Years										
1 In cell <b>M24</b> , use vlookup() to find out the total number of residents in Singapore between 30-34 years old in 2016.										
2 In cell M22, use hlookup() to find out the total number of residents in Singapore between 5-9 years old in 2011.										
3 In cells <b>C19</b> to <b>L19</b> , calculate the number of residents who are between the ages of 0 and 19 years old for each year.										
In cells N4 to N	<b>117</b> , calcula	ite the aver	rage numbe	er of resider	nts for each	age group	between 2	010 and 20	)19 inclusiv	e.
In cells C4 to L4, use conditional formatting to highlight years where the total residents between 0 and 4 years old was										

## **Total Residents in Singapore**

## Open the file Total Residents (2010-2019).xlsx

### Save the file as Total Residents (2010-2019)\_INDEX\_NAME.xlsx.

- 1 In cell M24, use vlookup() to find out the total number of residents in Singapore between 30-34 years old in 2016. [2]
- 2 In cell M22, use hlookup() to find out the total number of residents in Singapore between 5-9 years old in 2011. [2]
- 3 In cells C19 to L19, calculate the number of residents who are between the ages of 0 and 19 years old for each year. [2]
- 4 In cells N4 to N17, calculate the average number of residents for each age group between 2010 and 2019 inclusive. [2]
- 5 In cells C4 to L4, use conditional formatting to highlight years where the total residents between 0 and 4 years old was below the average calculated in cell N4.

Save and close your file.

[2]

Task 2

A hangman game is played by allowing the user to guess a word, one character at a time. A version of the game is implemented in the code shown below.

```
import random
word list = ['cup', 'milk', 'car', 'bus', 'bird', 'song']
chosen_word = word_list[random.randint(0,len(word_list))]
#print(chosen word)
grid = []
for i in range(len(chosen word)):
    grid.append(" ")
while " " in grid:
    print(grid)
    user = input("What's your guess? : ")
    if user in chosen word:
        print("{} is in the word!".format(user))
        grid[chosen word.index(user)] = user
    else:
        print("not found")
        print("try again")
    print()
print("Congratulations!")
```

Open the file HANGMAN.py

Save the file as HANGMAN\_INDEX\_NAME.py

6 Edit the program so that it:

(a)	Outputs a welcome message when the program is run.	[1]
(b)	Counts the number of rounds the player took to guess the word and output the number when the game ends.	[4]
(c)	Converts inputs to the lower case.	[2]
(d)	Tracks and outputs the guesses that have been entered every round.	[3]
~		

Save your program.

The code below shows the Python code for a number guessing game. The game follows the following rules:

- The secret number is between 1 and 999, inclusive.
- Only numerical inputs are accepted.
- If the player's guess is higher, the program will output that the guess is higher.
- If the player's guess is lower, the program will output that the guess is lower.
- The game ends when the user guesses the secret number correctly.

```
import randomn
NUM = random.randint0,999)
user = -1
while user != NUM:
   while True:
        user = input ("key in a number: ")
        if user.isdigit()
            user = user
            break
        else:
            print("numbers only.)
    if user = NUM:
        print("You've got it!")
    elif user <= NUM:
        print("Your guess is lower!")
    elif user < NUM:
        print("Your guess is higher!")
```

Open the file NUMBER.py

7

Save the file as NUMBER\_INDEX\_NAME.py

Identify and correct the errors in the program so that it works according to the rules given.

Save your program.

[10]

#### Task 4

A contact list is used to store the name, contact number, age and gender of a group of friends.

**8** Write a program that allows the user to key in the following information while adhering to the restrictions.

Information	Restrictions		
Name	Nil		
Contact Number	Must contain 8 digits.		
Age	Nil		
Gender	"M" or "F"		

For each **contact**, all 4 pieces of information must be keyed in.

After keying in all 4 pieces of information, output the information for each contact as shown below.

```
Contact list

Please key in the following pieces of information

Name: Tiny Tots

Contact Number: 98765432

Age: 5

Gender (M/F):F

Tiny Tots,98765432,5,F
```

Save the file as CLIST\_INDEX\_NAME.py

[10]

**9** When your program is complete, key in the following information and take a screen shot of each entry.

# • Test 01

Filename: Test 01 INDEX NAME

Name	Tiny Tots		
Contact Number	98765432		
Age	5		
Gender	F		

• Test 02

Filename: Test\_02\_INDEX\_NAME

Name	Looney Tunes		
Contact Number	1234567, 12345678		
Age	9		
Gender	Х, М		

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

[2]

- 10 Extend your code in **Q8** to:
  - Capitalise the first letter of the name
  - Restrict the age to between 0 and 150, inclusive
  - Check that the first digit of the phone number is either a 6, 8 or 9

Save the file as CLIST2\_INDEX\_NAME.py

11 Modify your code to output the name of the youngest person in the contact list.

You may use the following data to test your program:

Name	Tiny Tots
Contact Number	98765432
Age	5
Gender	F

Name	Looney Tunes
Contact Number	12345678
Age	9
Gender	M

Name	Elderberry			
Contact Number	61234757			
Age	99			
Gender	M			

Save the file as CLIST3\_INDEX\_NAME.py

[2]

[6]