Name: _____

Class: _____

Clementi Town Secondary School Preliminary Examination 2019 Secondary 4 Express



Computing

7155/02 28 August 2019 2 hours 30 mins

Register Number: _____

Δdditi	onal	Ma	toria	ale		Flor	otroi	nic	vore	ior	م م	MEM		c1	c (C 7 1	7 T N	C	VTC	eh v	ito f	filo		
CLEMENTI	TOWN S	SECON	DARY	SCHOO	DL CLE	EMENT	I TOWN	I SECO	ONDAR	Y SCł	HOOL	CLEMEN	TI TOW	N SE	COND	DARY	SCHO	DL CI	LEMENTI	TOWN	SECC	NDARY	SCHOO	L
CLEMENTI	TOWN S	SECON	DARY	SCHOO	DL CLE	EMENT	I TOWN	I SECO	ONDAR	Y SCł	HOOL	CLEMEN	TI TOW	N SE	COND	DARY	SCHO	DL CI	LEMENTI	TOWN	SECC	NDARY	SCHOO	L
CLEMENTI 7	TOWN S	SECON	DARY	SCHOC	DL CLE	EMENT	I TOWN	I SECO	ONDAR	Y SCł	HOOL	CLEMEN	TI TOW	N SE	COND	DARY	SCHO	DL CI	LEMENTI	TOWN	SECC	NDARY	SCHOO	L
CLEMENTI 7	TOWN S	SECON	DARY	SCHOC	DL CLE	EMENT	I TOWN	I SECO	ONDAR	Y SCł	HOOL	CLEMEN	TI TOW	N SE	COND	DARY	SCHO	DL CI	LEMENTI	TOWN	SECC	NDARY	SCHOO	L
CLEMENTI	TOWN S	SECON	DARY	SCHOC	DL CLE	EMENT	I TOWN	I SECO	ONDAR	Y SCł	HOOL	CLEMEN	ΓΙ ΤΟΨ	N SE	COND	DARY	SCHO	DL CI	LEMENTI	TOWN	SECC	ONDARY	SCHOO	L

Additional Materials: Electronic version of MEMBERS'S_SAVING.XLSX date file Electronic version of SCORE.PY file Electronic version of BMI.PY file Data Resource Files Quick reference for Python sheet

READ THESE INSTRUCTIONS FIRST

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.

For Examin	er's Use
	50

This question paper consists of $\underline{7}$ printed pages, including this cover page.

2 Secondary 4 Express

Answer all the questions.

Task 1

A bank uses spreadsheet software to record the amount and duration that members save in their fixed deposit account. The bank offers compounded interest to all its savers. You are required to finish setting up the spreadsheet to record the interest they will be getting.

Open the file MEMBER'S_Savings. You will see the following data.

	A	в	С	D	Е	F
1			Saver's List			
2						
		Amount	Number of		Interest	Bonus
3	Saver number	saved	Years	Rate	Earned	Eligibility
4	M0001	\$1,000.00	2			
5	M0002	\$4,500.00	3			
6	M0003	\$12,000.00	8			
7	M0004	\$19,000.00	7			
8	M0005	\$4,000.00	9			
э	M0006	\$3,800.00	2			
10	M0008	\$450.00	1			
11	M0009	\$1,000.00	4			
12	M0010	\$9.90	1			
13	M0011	\$990.00	5			
14	M0012	\$1,800.00	7			
15	M0014	\$6,700.00	2			
16	M0015	\$45,654.00	4			
17	M0016	\$12,344.00	1			
18	M0017	\$8,900.00	3			
19	M0018	\$5,391.00	3			
20	M0019	\$1,200.00	3			
21						
22	Total savings					
23	Jumber of members					
24	Longest Duration					
25	_				Interest Rate	s
26				Number of	Description	Rate
27				1	Onellear	1%
					Two to	
28				2	three years	2%
29				4	Four to six years	2%
30				7	Seven years and	3%
- 24						

Save the file as **MEMBER'S_Savings_**<your_name>_<class>_<index number>.

1	In cell B23 enter a formula to calculate the total amount of savings.	[1]
2	In cell B24 enter a formula to count how many members there are.	[1]
3	In cell B25 enter a formula to count the longest duration.	[1]
4	Use an appropriate function to search for the Rate in the Interest Rates table and use it to complete the Rate column.	[2]

- 5 Enter a formula to calculate the interest members will be earning and use it to [3] complete the **Interest Earned** column.
- 6 Use a conditional statement to identify those savers who are eligible for bonus. [2] They will have to save more than \$2000 and over a period more than THREE years. Put a YES in the Bonus Eligibility column. Otherwise, put NO in the Bonus Eligibility column.

Save and close your file.

Task 2 begins on the next page.

Computing

Task 2

The following program accepts scores for TEN students and prints out the lowest in score and the highest in score. The program should accept values from 0 to 100 marks.

```
highest = 0
lowest = 1000
students = 10
for count in range(students):
    score = int(input("Enter a score for the student: "))
    if score > highest:
        highest = score
    if score < lowest:
        lowest = score
print ("The highest score is ", highest)
print ("The lowest score is ", lowest)
```

Open the file **SCORE.py**.

Save the file as SCORE_<your name>_<class>_<index number>

7	Edit	Edit the program so that it:									
	(a)	Accepts score for only 15 students.	[1]								
	(b)	Prints out the average score of the 15 students.	[3]								
	(c)	Tests if the score is between 0 to 100 marks, and if not, asks the user for input again with a suitable prompt.	[4]								
	Sav	e your program.									
8	Sav	e the file as VARSCORE_ <your name="">_<class>_<index number=""></index></class></your>	[2]								
	Edit Save	Edit your program so that it works for any number of students. Save your program.									

Task 3 begin on the next page.

Task 3

In a nation-wide survey, the BMI (Body Mass Index) of all students are being measured. The formula for BMI is as follows:

$BMI = \frac{Mass (in kilograms)}{Height^2 (in metres)}$

The classification of BMI is in the table below.

Health Risk	BMI (kg/m ²)		
Risk of developing problems such as nutritional deficiency	under 18.5		
and osteoporosis.			
Low Risk (healthy range).	18.5 to under 23		
Moderate risk of developing heart disease, high blood	23 to 27.5		
pressure, stroke and diabetes.			
High risk of developing heart disease, high blood pressure,	over 27.5		
stroke and diabetes.			

Students with BMI under 18.5 or over 27.5 are classified as "at risk". A program is used to calculate and store the values of "at risk" BMI.

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

```
risk = \{\}
count = 0
bmi total = 0
while counter <= 10:
    height = int(input("Please enter your height in
meters: "))
    weight = int(input("Please enter your weight in kg:
"))
    bmi = weight/height*height
    bmi total += bmi
    if bmi > 18.5 or bmi < 27.5:
        print("You are at risk.")
        risk = risk + bmi
    counter += 1
bmi average = bmi total/counter
print("Average BMI "+ bmi average)
print("The following students with these BMI are at
risk.")
for i in risk:
    print(i)
Open the file BMI.py
```

Save your program as BMI_<Your name>_<Class>_<Index Number>

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

[Turn over

Task 4

CT Generation is a handphone application used to deliver discounts to household bills. You are a member of the app design team responsible for the final step - calculate the discount entitlement of each family.

To register, each household is to enter the year of birth for all its family members. The app will receive an input in the following format:

Eg 1. 1965, 1970, 2004, 2007

Every household will be entitled to 5% for each member above 55 years old, and 1% for each member below 16 years old.

Using the program CT_Generation(), the program should give the following output.

Input	Output						
1965, 1970, 2004, 2007	There are 4 members. There is/are 0 members above 55 years old. There is/are 2 members below 16 years old.						
	You are entitled to 2% discount.						
1950, 1955, 1979.	There are 3 members. There is/are 2 members above 55 years old. There is/are members below 16 years old.						
	You are entitled to 10% discount.						

Table 1

Your program should be able to perform the following with necessary validation:

- (a) Enter data in the format a,b,c,d, e, ... where a refers to the year of birth.
- (b) Only allow data entry of numbers for member in the household.
- (c) Find the total number of family members in each household.
- (d) Calculate and display the total discount entitlement.
- (e) Your output must look like the output in **Table 1**.
- **10** Write your program and test that it works.

Save your program as CT_Mart_<Your name>_<Class>_<Index Number>

11 When your program is complete, using the inputs in Table 1, take a [2] screenshot and save as: CHECK_<your name>_<class>_<index number> Save your files in either .png or .jpg format.
12 Save your program as CT_Generationa_Gender(). [8] Extend your program to

(a) accept "M" or "F" behind the year and output the number of male

and female in the household.

- (b) print out the oldest and youngest ages.
- (c) print out the average age in the household, rounded to 1 decimal place.

Input	Output
1970M, 1967F, 2007M,	There are 4 members.
2004M	There is/are 0 members
	above 55 years old.
	There is/are 2 members
	below 16 years old.
	There are 2 Male and 2
	Female.
	The oldest is 52 years old
	and the youngest is 12
	years old.
	The average age is 32
	years old
	You are entitled to
	2% discount.
1950M, 1955F, 1979M.	There are 3 members.
	There is/are 2 members
	above 55 years old.
	There is/are members
	below 16 years old.
	There are 2 Male and 1
	Female.
	The oldest is 69 years old
	and the youngest is 40
	years old.
	The average age is 57.7
	years old.
	You are entitled to
	10% discount.

CHECK_Gender<your name>_<class>_<index number>

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

Computing

~ END OF PAPER ~