Name: ______________________ **Class:** ______ **Index Number:** _______

THE BEST IS YET TO BE

Anglo-Chinese School (Barker Road)

PRELIMINARY EXAMINATION 2022

SECONDARY FOUR EXPRESS

COMPUTING PAPER 2

7155/02

2 HOUR 30 MINUTES

INSTRUCTIONS TO CANDIDATES

Additional Materials:	Electronic version of CUSTOMERS.XLSX data file Electronic version of FIB.PY Python file Electronic version of DATE.PY Python file Electronic version of GET_PRICE.PY Python file Insert Quick Reference Glossary

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.

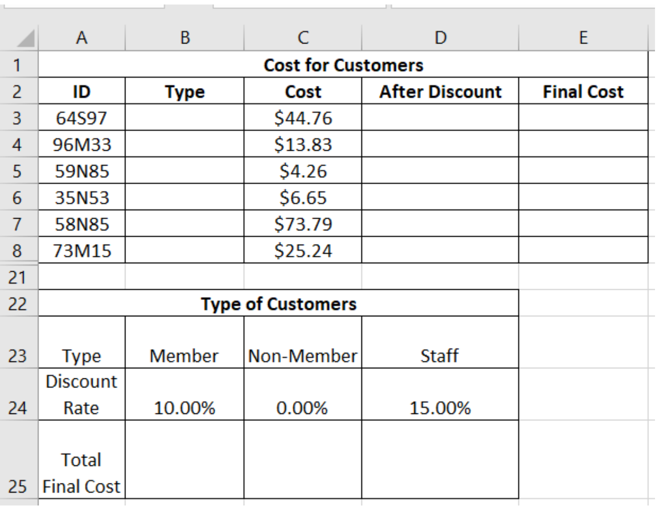
*This document consists of **7** printed pages, inclusive of the cover page*

Task 1

The staff uses spreadsheet software to calculate the final cost that the customers incurred in the shop yesterday.

You are required to finish setting up the spreadsheet.

Open the file **CUSTOMERS.xlsx**. You will see the following data. Only selected rows from the file have been shown below.



	A	B	C	D	E
1	Cost for Customers				
2	ID	Type	Cost	After Discount	Final Cost
3	64S97		$44.76		
4	96M33		$13.83		
5	59N85		$4.26		
6	35N53		$6.65		
7	58N85		$73.79		
8	73M15		$25.24		
21					
22	Type of Customers				
23	Type	Member	Non-Member	Staff	
24	Discount Rate	10.00%	0.00%	15.00%	
25	Total Final Cost				

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

1 Use a function to extract the letter that is between the numbers to complete the **Type** column in the **Cost for Customers** table. Display “Member” for “M”, “Non-Member” for “N” and “Staff” for “S”.

[2]

2 Use an appropriate function to search for the **Discount Rate** in the **Type of Customers** table and use it to complete the **After Discount** column. The after discount cost must take into the consideration of the cost that each customer incurred.

[2]

3 Use a function to round down the after discount cost to the nearest dollar to complete the **Final Cost** column.

[2]

4 In cell **B25**, **C25** and **D25** use a function to calculate the total final cost for each type of customer.

[2]

5 In cells **A3** to **E20** use a formatting tool to change the colour of the row to yellow for staff type.

[2]

Save and close your file.

Task 2

A Fibonacci sequence is the integer sequence of 0, 1, 1, 2, 3, 5, 8....

The first two terms are 0 and 1. All other terms are obtained by adding the preceding two terms.

The following program outputs the first five terms in the Fibonacci sequence.

```
n1 = 0
n2 = 1

nterms = 5

for i in range(nterms):
    print(n1)
    nth = n1 + n2
    n1 = n2
    n2 = nth
```

Open the file **FIB.py**

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

6 Edit the program so that it works for any number of terms. The program must display a suitable input message.

[1]

7 Edit the program to only accept a positive integer to be input. A suitable error message must be displayed if the nterms is not in the range. The program must loop until a valid nterms is input.

[3]

8 Edit the program to store the Fibonacci sequence in a list. Display the list at the end of program.

[3]

Save your program.

9 Save your program as **INFIB_<your name>_<class>_<index number>.**

Edit the program to allow user to input another positive integer and display if the integer is in the first hundredth terms of the Fibonacci sequence. You do not need to validate the input.

[3]

Save your program.

Task 3

The following program allows a user to input a date from 1900 to the current year, then validate the date until it is valid. Note that the current year now is 2022.

The program:

- asks the user to input a date in the format of DD-MM-YYYY
- display appropriate warning messages when the input is invalid
- allows the user to re-input the date until the date is valid

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

```
while True:
date = input("Enter the date (DD-MM-YYYY): ")
    test = date
    if len(test)= 10 and test[2]=="-" and test[5]=="-":
        day = int(test[0:2])
        month = int(test[3:])
        year = int(test[6:])
        check_year = year>1900 and year<=2000
        check_month = month>=1 or month<=12
        check_day_31 = day<=31 and (month in [1,3,5,7,8,10,12])
        check_day_30 = day<=31 and (month in [4,6,9,11])
        check_day_Feb = month == 0 and ((day<=29 and year%4==0) or day<=28)
        if check year:
            if check_month:
                if check_day_31 or check_day_30 or check_day_Feb
                    break
                else:
                    print("Error in day")
            else:
                print("Error in year")
        else:
            print("Error in month")
    else:
        print(Error in format")
print("Date accepted")
```

Open the file **DATE.py**

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

10 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 create a program for a neighbourhood cake shop to get order from customers.

Open the file **GET_PRICE.py**. You will see the following function which takes in the cake option and return the corresponding price for that particular cake. **DO NOT MODIFY OR CHANGE THIS FUNCTION.**

```
def get_price(option):
    cake_list = ["A", "B", "C", "D","E","F","G","H"]
    price_list = [25, 22, 38, 35, 15, 40, 53, 20]
    position = cake_list.index(option)
    return price_list[position]
```

Sample executions:

```
>>> get_price("A")
25

>>> get_price("E")
15
```

11 Save your program as **GET_INPUT_**<your name>**_**<class>**_**<index number>**.py**

In the same program, write another function `get_input()` which generates the choice of cake of the customer. Make sure only one single uppercase letter from A to H is accepted. A suitable error message must be displayed. The program must loop until a valid letter is input.

[3]

Sample executions:

```
>>> get_input()
Enter the choice of cake: abc
Enter an uppercase letter between A to H only
Enter the choice of cake: a
Enter an uppercase letter between A to H only
Enter the choice of cake: A
'A'

>>> get_input()
Enter the choice of cake:
Enter an uppercase letter between A to H only
Enter the choice of cake: 1
Enter an uppercase letter between A to H only
Enter the choice of cake: G
'G'
```

Save your program.

12 Save your program as **GET_ORDER_**<your name>**_**<class>**_**<index number>**.py**

In the same program, write another function `get_order()` which displays the total amount that customer needs to pay.

Use the `get_input()` function to get the choice of cake.

Use the `get_price()` function to get the price of the cake.

The function must loop until no more additional cake is ordered.

Display the subtotal, which is the total amount of all the cake(s) ordered.

Display the GST amount, which is 7% of the subtotal. Round the GST amount to 2 decimal places. If the cent is more than $0.05, round the amount down to the nearest $0.05. If the cent is less than $0.05, round the amount down to the nearest $0.00.

Display the final total, which is the sum of subtotal and GST amount.

Make sure the display of all the outputs must be the same as the sample executions.

[14]

Sample executions:

```
>>> get_order()
Enter the choice of cake: A
More purchase? Y or N: Y
Enter the choice of cake: G
More purchase? Y or N: N

Subtotal       $ 78
GST            $ 5.45
Total          $ 83.45

Thank you!

>>> get_order()
Enter the choice of cake: G
More purchase? Y or N: N

Subtotal       $ 53
GST            $ 3.70
Total          $ 56.70

Thank you!
```

Save your program.

13 Save your program as **GET_ORDER2_**<your name>**_**<class>**_**<index number>**.py**

In the same program, write another function `get_order2()` which displays the total amount that customer needs to pay.

Assume that each type of cake has quantity of 2 cakes per day.

Check if the type of cake is still available for order. If it is not, display a warning message.

[3]

Sample executions:

```
>>> get_order2()
Enter the choice of cake: A
Another purchase? Y or N: Y
Enter the choice of cake: A
Another purchase? Y or N: Y
Enter the choice of cake: A
The cake is not available
Another purchase? Y or N: N

Subtotal       $ 50
GST            $ 3.55
Total          $ 53.55

Thank you!
```

Save your program.

End of Paper