ZHONGHUA SECONDARY SCHOOL

PRELIMINARY EXAMINATION 2022
SECONDARY 4 EXPRESS

Candidate's Name	Class	Register Number

COMPUTING 7155/02

Paper 2 Practical

13 September 2022
2 hours 30 minutes

Additional Materials – Digital versions of:
GOLDSTRIKEINTERNALMATCH.xlsx
WEIGHFISH.py
MULTIPLY_BY_RA.py
4DGUESS.py

READ THESE INSTRUCTIONS FIRST

Write your name, index number and class in the spaces at the top of this page.
Write in dark blue or black pen.
You may use an HB pencil for any diagrams, graphs or rough working.
Do not use staples, paper clips, glue, correction fluid or correction tape.

Approved calculators are allowed.

Answer **all** questions.

The number of marks is given in brackets [ ] at the end of each question or part question.
You should show all your working.
The total of the marks for this paper is **50**.

Rename the folder **A_Student** to **<Index_Number>_<Your_Class>_<Your Name>**
Store all your answer files in the renamed folder.

For Examiner's Use	
TOTAL	

Setter: Mr. Calvin Heng
Vetted by: Mr. Low Kee Ley

Task 1

The Gold Strike archery club is preparing to select members for an upcoming international competition, by using a club level competition.

The club uses spreadsheet software to record the details of the members scores.

You are required to finish setting up the spreadsheet to record details of the members scores.

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

	A	B	C	D	E
1	**GOLD STRIKE ARCHERY CLUB MEMBERS**				
2					
3					
4	**Name**	**Target 80**	**Target 40**	**Score**	**Ranking**
5	Arnold	202	190		
6	Bill	184	143		
7	Charles	193	112		
8	Denver	215	175		
9	Edward	116	90		
10	Felix	240	187		
11	George	135	88		
12	Henry	148	89		
13	Ichtyus	235	192		
14	James	202	167		
15	Kenward	200	100		
16	Leonard	154	85		
17	Mason	121	10		
18	Nixon	211	154		
19	Oscar	192	67		
20					
21					
22	**Highest Score**				
23	**Average Score**				

Save the file as **GOLDSTRIKESELECTION_2022_**<your name>_<index number>**.xlsx**

1 In column **D** enter a formula that uses a function to add up the scores for Target 80 and Target 40 for each member. [1]

2 In cell **B22** enter a formula that uses a function to find the highest score. [1]

3 In cell **B23** enter a formula that uses functions to find the average score of all the members. The average score must be rounded up to the nearest integer. [2]

4 In the **Ranking** column enter a formula that uses functions to:

[4]

- Find the highest score member and displays the text **Gold** for that member
- Find the second highest score member and display the text **Silver** for that member.

All other cells in the column must be empty.

5 In cells **A5** to **E19** use a conditional formatting tool to make the text red in the row that contains the lowest scoring member. [2]

Task 2

A program is used to input the weight of a fish on a fish farm. The minimum weight for the fish to be sold on the market is 1.2 kg. The following program checks whether the fish meets the minimum weight requirement. The program allows the farmer to enter the weights of 5 fishes. It will display a message after each weight entry is made, to indicate if the fish is ready to be sold on the market or not.

```
number_fish = 5
minimum_weight = 1.2
for i in range(number_fish):
    weight = float(input("Weight of fish: "))
    if weight >= minimum_weight:
        print("Fish weight meets requirement.")
    else:
        print("Fish weight does not meet requirement.")
```

Open the file **WEIGHFISH.py**

Save the file as **MARKETFISH_2022_**<your name>_<index number>**.py**

6 Edit the program so that it will work for any number of fish. The program must display a suitable input message.

Save your program. [2]

7 In order to meet conservation and sustainability requirements set by the Singapore Food Agency (SFA), farmed fish must be greater than 80 cm in length to be market ready.

Edit the program so that it checks that a fish meets both weight and length requirements. The program must display:

- "This fish is not heavy enough. Return to Extra Feed Pen.", if the fish meets length requirements but not weight requirements.
- "This fish is not long enough. Return to Growing Pen.", if the fish meets weight requirements but not length requirements.
- "This fish is not heavy enough and not long enough. Return to Original Pen.", if the fish does not meet weight and length requirements.
- "This fish is heavy enough and long enough. Bag it for Market.", if the fish meets weight and length requirements.

Save your program.

[4]

8 For statistical reporting to SFA, edit the program so that it stores the weight of the marketable fish in a list (name the list as fishweight, and the length of the marketable fish in a list (name the list as fishlength).

Output the average weight of all the fish in the list.

Output the average length of all the fish in the list.

Save your program as
AVG_FISH_WT_HT_2022_<Your_Name>_<Index_Number>.**py**

[4]

Task 3

A program performs multiplication using the repeated addition algorithm.

The program asks the user for two integers. The user can enter two positive integers, or a positive and negative integer, or two negative integers.

The program uses the second integer as the multiplier and using the repeated addition algorithm, produces the multiplied result with the correct sign (positive or negative).

Open the file **MULTIPLY_BY_RA.py**. You will see the following program.

```
def mult(a, b):
    # multiply function using repeated addition algorithm
    # a and b must be integers, by using type casting
    a = int(a)
    b = str(b)

    # validate if either a or b is zero
    if a = 0 and b == 0:
        return 0

    # convert both integers to absolute value
    a , b = abs(a) , abs(b)

    # check sign
    if a < 0 and b < 0: neg_sign_flag = True
    elif a < 0 or b < 0: neg_sign_flag = True
    else: sign_flag = False

    # iterate addition
    accum = 1
    for i in range(a): # make b the multiplier
        accum += a

    # return the multiplied value, with the appropriate sign
    if neg_sign_flag:
        return accum
    else:
        return accum

# User Input and Test Run
a = input("please input the first integer: ")
b = input("please input the second integer: ")
print(str(a)+" multiplied by "+str(b)+" equals "+str(multiply(a,b)))
```

Save the file as **MULTIPLY_CORRECTLY_2022_**<your name>_<index number>**.py**

9 Identify **and** correct the errors in the program so that it works according to the requirements given. Save your program. [10]

Task 4

Base Task

10 You are required to write a number guessing game.

The game begins with the computer generating a four-digit number randomly.

The player inputs a four-digit number. There is no requirement for input validation.

If the guess is correct, the program will congratulate the player.

If the guess is incorrect, the program will only inform the player if there are correctly guessed digits and in the correct position. All other scenarios will result in a standard message. Please follow this table for specific output messages:

Scenario	**Message**
Correct guess on first try	Great! You guessed the number in 1 try!
Totally incorrect guess e.g. Number is 5660, Player guessed 1234	None of the numbers in your input match!
1 digit correctly guessed e.g. Number is 5660, Player guessed 5432	You only got 1 digit(s) correct These numbers in your input were correct: 5 X X X
2 digits correctly guessed e.g. Number is 5660, Player guessed 5467	You only got 2 digit(s) correct These numbers in your input were correct: 5 X 6 X
3 digits correctly guessed e.g. Number is 5660, Player guessed 5260	You only got 3 digit(s) correct These numbers in your input were correct: 5 X 6 0
Player makes correct guess after fifth try.	You guessed the number correctly! It took you 5 tries!

Note: correct digit but incorrect position is treated as an incorrect guess.

11 Open the file **4DGUESS.py** The opening lines on generating the 4 digit number have been written for you. The *initial testing only* line of code is to aid in your program development. Please keep this line of code for submission.

Save your file as **4DGUESS_<Your Name>_<Index number>.py**

12 Your program must allow the player to input a 4 digit number. [2]

13 Your program must be able to address all the scenarios listed in the table above. [10]

14 Test your program with the following test cases:

A Correct guess on first try [1]

B Totally incorrect guess [1]

C 1 digit, 2 digit, 3 digit and finally correct guess [2]

Provide a screen print of the above test flow. Save the screen print as a bitmap with the filename:

4DGUESSTESTING_<Your Name>_<Index Number>.jpg [1]

Extension Task

15 Save the program as **5DGUESS_<Your Name>_<Index Number>.py**

You will increase the difficulty of the game by getting the player to guess a 5 digit number.

Take note that you will need to make changes in the following areas:

- Random number generation.
- Player Input, no requirement for input validation.
- Game Loop. [3]

~ End of Paper ~