ASSIGNMENT 2

PROGRAMMING TECHNIQUE 1

SEM 1, 2020/2021

INSTRUCTIONS TO THE STUDENTS

- *This assignment must be done **in pairs** (group consisting of 2 members).*
- *Please refer to the group list to find out your group members.*
- *Your programs must follow the input and output as required in the text and shown in the examples. You must test the programs with (but not limited to) all the input given in the examples.*
- *Any form of plagiarisms is **NOT ALLOWED**. Students who copied other student's program/ assignment will get **ZERO** mark (both parties, student who copied and student that share their work).*
- *Please insert your **name and partner's name, matrics number and date** as a comment in your program.*

SUBMISSION PROCEDURE

- *Please submit this assignment no later than **December 27, 2020, Sunday (00:00 MYT)**.*
- *Only one submission per pairs (group) that includes two files are required for the submission which is the source codes (the file with the extension .cpp).*
- *Submit the assignment via the UTM's e-learning system.*

QUESTION 1 (Pre-defined functions)

Joe's Tile Depot, a tile shop in Taman Universiti, needs a program to calculate the number of tiles required by its customers. The program should perform the following steps:

a) Ask the user for the code of tile the customer wants to order as well as the price of the tile. The tile's code is a four-digit integer number, where the first two digits indicates the tile's width, while the next two digits is the tile's length. Both dimensions are in inches. For example, the tile code **2412** means that the tile is **24** inches in width and **12** inches in length. The tile's price is in RM.

b) Ask the user for the width and length, in feet, of surface the customer wants to tile. The program should convert the metric of both dimensions into inches and store them as floating point numbers. Note that 1 foot is equivalent to 12 inches.

c) Determine the width and length of the tile from its code.

d) Calculate the number of tiles required. This figure can be obtained by dividing the size of the surface to the size of the tile. As tiles are sold in pieces, the program should round the figure up to the nearest integer. Table 2 shows some examples:

Table 2: Examples of the number of tiles required by round up to the nearest integer

Estimate number of tiles	Actual number of tiles required
202.9	203
101.1	102
98.5	99
88.0	88

Notes: Requirements (c) and (d) can be achieved by using only simple arithmetic operations.

e) Display the output which contains the following information:
   i) The code, width, length and price of the tile.
   ii) The number of tiles required.
   iii) The amount that the customer need to pay.

The example runs of the program is shown in **Figure 1**. ***Note:*** The **bold texts** indicate input from the user.

```
Enter the tile's code => 2412
Enter the tile's price (RM) => 12.15
Enter the surface's width (in feet) => 12.5
Enter the surface's length (in feet)=> 11.9

Customer's Receipt
==========
The tile to purchase:
Code : 2412
Width : 24 inches
Length: 12 inches
Price : RM 12.15 per piece

The number of tiles required: 75 pieces
The total cost: RM 911.25
```

(a) Example Run 1

```
Enter the tile's code => 0804
Enter the tile's price (RM) => 1.20
Enter the surface's width (in feet) => 9.5
Enter the surface's length (in feet)=> 5.5

Customer's Receipt
==========
The tile to purchase:
Code : 804
Width : 8 inches
Length: 4 inches
Price : RM 1.20 per piece

The number of tiles required: 236 pieces
The total cost: RM 283.20
```

(b) Example Run 2

Figure 1: Example Runs

QUESTION 2 (User-defined Functions)

The Body Mass Index (BMI) is a quick and easy screening method to determine the weight status of a person. The BMI of a person is calculated as:

$$BMI = \frac{weight}{height^2}$$

where, *weight* and *height* are measured in kilograms and meters, respectively. A person is considered to be normal, if his or her BMI is between 18.5 and 25. If the BMI is less than 18.5, the person is considered to be underweight. If it is greater than 25 but not more than 30, the person is considered to be overweight. If the index is above 30, the person is considered to be obese.

Based on the information of BMI described above, write a C++ program to determine the weight status of a list of person. The program should fulfil the following requirements:

f) The user is required to enter the name, weight and height of each person in the list. The program should provide a mechanism to control the loop of reading input. For example, the program will keep reading input until the user enters an empty name. Another example is that the user is firstly asked for the number of person he or she wants to enter.

g) The output of the program contains the following information:
- The list of persons along with their names, weights, heights, BMIs, and weight status.
- The overall BMI and weight status. The overall BMI is calculated as:

$$OverallBMI = \frac{OverallWeight}{OverallHeight^2}$$

where, *OverallWeight* and *OverallHeight* are obtained as the average weight and height over all persons in the list, respectively. Then the overall status should be determined based on the overall BMI.

h) The program should provide the following functions:
- **`getBMI`** to calculate BMI.
- **`getStatus`** to determine the weight status based on BMI.

Note: You may want to define other functions if necessary.

Figure 2 shows some example runs of the program with their set of user inputs and corresponding output. ***Notes:*** As for example of the program, the mechanism used for controlling the loop of reading user inputs is that it ***keeps reading input until the user enters an empty name***.

```
Enter name or press <ENTER> key to end=> Hamid
Enter weight(kg) and height(m) => 76.8  1.6

Enter name or press <ENTER> key to end=> Elias
Enter weight(kg) and height(m)  => 64  1.6

Enter name or press <ENTER> key to end=> Amaleena
Enter weight(kg) and height(m) => 18.5  1

Enter name or press <ENTER> key to end=>
```

(a) User Inputs of Run 1

```
Name   : Hamid
Weight: 76.80 kilograms
Height: 1.60 meters
BMI    : 30.00
Status: Overweight

Name   : Elias
```

```
Weight: 64.00 kilograms
Height: 1.60 meters
BMI   : 25.00
Status: Normal

Name  : Amaleena
Weight: 18.50 kilograms
Height: 1.00 meters
BMI   : 18.50
Status: Normal

Overall BMI : 27.09
Overall Status : Overweight
```

(b) Output of Run 1

```
Enter name or press <ENTER> key to end=> Ali
Enter weight(kg) and height(m) => 74.6  1.71

Enter name or press <ENTER> key to end=> Bakar
Enter weight(kg) and height(m)  => 61  1.68

Enter name or press <ENTER> key to end=> Daud
Enter weight(kg) and height(m)  => 66.4  1.89

Enter name or press <ENTER> key to end=>
```

(c) User Inputs of Run 2

```
Name  : Ali
Weight: 74.60 kilograms
Height: 1.71 meters
BMI   : 25.51
Status: Overweight

Name  : Bakar
Weight: 61.00 kilograms
Height: 1.68 meters
BMI   : 21.61
Status: Normal

Name  : Daud
Weight: 66.40 kilograms
Height: 1.89 meters
BMI   : 18.59
Status: Normal

Overall BMI : 21.74
Overall Status : Normal
```

(d) Output of Run 2

Figure 2: Example Runs