ASSIGNMENT 1
PROGRAMMING TECHNIQUE I (SECJ/SCSJ 1013)

INSTRUCTIONS TO THE STUDENTS:

- *This assignment must be done* ***<u>in pairs</u>*** *(please refer to the list of groups).*
- *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, IC Number, section of your class and date*** *as a comment in your program.*

SUBMISSION PROCEDURE:

- *Please submit this assignment no later than* **November 14, 2019, Thursday (5.00 pm)**.
- *Only one submission per pairs (partners) that includes three files are required for the submission which is the source code (the file with the extension .cpp).*
- *Submit the assignment via the UTM's e-learning system.*

QUESTION 1

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 program to determine the weight status of a list of person. The program should fulfill the following requirements:

Input:

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. (***Note:*** In the Figure 1, the values in **bold** are input by the user)

Output:

The output should contain the following information:

a. The list of persons along with their names, weights, heights, BMIs, and weight status.

b. 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.

User-defined Functions:

The program should provide the following functions:

a. **`getBMI`** to calculate BMI.

b. **`dispStatus`** to display the weight status based on BMI.

Note: You can also define other function(s) if necessary.

Example Run

```
Enter name or press <ENTER> key to end => Ali
Enter weight(kg) and height(m) => 85 1.7

Name  : Ali
Weight: 85.00 kilograms
Height: 1.70 meters
BMI   : 29.41
Status: Overweight

Enter name or press <ENTER> key to end => Alia
Enter weight(kg) and height(m) => 56 1.54

Name  : Alia
Weight: 56.00 kilograms
Height: 1.54 meters
BMI   : 23.61
Status: Normal

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


Overall BMI : 26.86
Overall Status : Overweight
```

Figure 1: Example output for Question 1

Note: The mechanism used for controlling the loop of reading user inputs in the example of output shown in Figure 1, is that *it keeps reading input until the user enters an empty name*.

QUESTION 2

A hospital supply company wants to market a program to assist with the calculation of intravenous rates. Design and implement a program that interacts with the user as follows.

Write a complete C++ program that helps the company to calculate intravenous rates. The program should perform the following steps. (***Note:*** that the values in *italics* are input by the user while the values in **bold** are the final result for each problem)

1. Provide a menu driven screen for user to select the problem choice.
2. If the user enters ***1***, the program should ask the user
    a) to enter the rate of intravenous drop in ml/hr
    b) to enter the intravenous tubing drop factor (drops/ml)

   From the given information, calculate and display the intravenous drop rate per minute. The screen displays for problem 1 is as in **Figure 2**.

```
INTRAVENOUS RATE ASSISTANT

Enter the number of the problem you wish to solve.
     GIVEN A MEDICAL ORDER IN            CALCULATE RATE IN
(1) ml/hr & tubing drop factor             drops/min
(2) mg/kg/hr & concentration in mg/ml      ml/hr
(3) QUIT

Problem => 1
Enter rate in ml/hr => 150
Enter tubing's drop factor(drops/ml) => 15
The drop rate per minute is 38.
```

Figure 2: Screen Display for Problem 1

3. If the user enters ***2***, the program should ask the user
    a) to enter input rate in mg/hr
    b) to enter the patient weight in kg
    c) to enter the concentration rate of the drug in mg/hr

   From the given information calculate and display rate in millilitres per hour. The screen display for problem 2 is as in **Figure 3**.

```
INTRAVENOUS RATE ASSISTANT

Enter the number of the problem you wish to solve.
     GIVEN A MEDICAL ORDER IN            CALCULATE RATE IN
(1) ml/hr & tubing drop factor             drops/min
(2) mg/kg/hr & concentration in mg/ml      ml/hr
(3) QUIT

Problem => 2
Enter rate in mg/hr => 0.6
Enter patient weight in kg => 70
Enter concentration in mg/ml => 1
The rate in millilitres per hour is 42.
```

Figure 3: Screen Display for Problem 2

4. If the user enters ***3***, the program will display the screen as in Figure 4.

```
INTRAVENOUS RATE ASSISTANT

Enter the number of the problem you wish to solve.
     GIVEN A MEDICAL ORDER IN        CALCULATE RATE IN
(1) ml/hr & tubing drop factor             drops/min
(2) mg/kg/hr & concentration in mg/ml      ml/hr
(3) QUIT

Problem => 3
You have chosen to quit the program.
Thank you for using our system.
```

Figure 4: Screen Display for Problem 3

5. If the user enters any other problem number besides 1, 2 or 3, the program will display the screen as in **Figure 5**.

```
INTRAVENOUS RATE ASSISTANT

Enter the number of the problem you wish to solve.
     GIVEN A MEDICAL ORDER IN        CALCULATE RATE IN
(1) ml/hr & tubing drop factor             drops/min
(2) mg/kg/hr & concentration in mg/ml      ml/hr
(3) QUIT

Problem => 6
Please run the system again and choose a problem number between 1
and 3.
```

Figure 5: Screen Display for Invalid Problem Choice

To assist the company in developing the program, you should implement the following functions:

`getProblem` – A function with no input parameters. It will display the user menu, then inputs and returns from the function the value of the problem number selected.

`getRateDropFactor` – This is a non-returning function which prompts the user to enter the data required for problem 1, and sends this data back to the calling module through the use of reference parameters.

`getKgRateConc` – This is also a non-returning function which prompts the user to enter the data required for problem 2, and sends this data back to the calling module through the use of reference parameters.

`figDropsMin` – This function takes rate and drop factor as input parameters and it returns the value of drops/min (rounded to the **nearest** whole drop, for example 23.2 to 24 and 23.7 to 24).

`byWeight` – This function takes as input parameters rate in mg/hr, patient weight in kg, and concentration of drug in mg/ml and it returns the value ml/hr (rounded, for example 26.2 to 26 and 26.6 to 27). *Note:* Formula to calculate rate in ml/hr = rate in mg/hr × weight in kg × concentration of drug in mg/ml.

Your program should also be written to show the implementation of a loop(s).

On the whole, the program execution for all problem choices will be depicted as follows:

```
INTRAVENOUS RATE ASSISTANT

Enter the number of the problem you wish to solve.
     GIVEN A MEDICAL ORDER IN            CALCULATE RATE IN
(1) ml/hr & tubing drop factor             drops/min
(2) mg/kg/hr & concentration in mg/ml      ml/hr
(3) QUIT
Problem => 1
Enter rate in ml/hr => 150
Enter tubing's drop factor(drops/ml) => 15
The drop rate per minute is 38.

Enter the number of the problem you wish to solve.
     GIVEN A MEDICAL ORDER IN            CALCULATE RATE IN
(1) ml/hr & tubing drop factor             drops/min
(2) mg/kg/hr & concentration in mg/ml      ml/hr
(3) QUIT

Problem => 2
Enter rate in mg/kg/hr => 0.6
Enter patient weight in kg => 70
Enter concentration in mg/ml => 1
The rate in millilitres per hour is 42.

Enter the number of the problem you wish to solve.
     GIVEN A MEDICAL ORDER IN            CALCULATE RATE IN
(1) ml/hr & tubing drop factor             drops/min
(2) mg/kg/hr & concentration in mg/ml      ml/hr
(3) QUIT
```

```
Problem => 3
You have chosen to quit the program.
Thank you for using our system.
```

```
For choices which are not in the range of 1 to 3 example -1 or 6:
```

```
Problem => 6
Please run the system again and choose a problem number between 1 and 5.
```

QUESTION 3

Write a complete C++ program for Question 12, page 94-95, Exercise 3, Lab 4 in Lab Module: Programming Techniques I C++. ***Note:*** You are only need to write the program without drawing the structured chart.