CONFIDENTIAL

SCHOOL OF COMPUTING
Faculty of Engineering

UNIVERSITI TEKNOLOGI MALAYSIA

SKILL-BASED TEST 1
SET A

SEMESTER II 2019/2020

SUBJECT CODE : SECJ/SCSJ 1023
SUBJECT NAME : PROGRAMMING TECHNIQUE II
YEAR/COURSE : 1 (SECB/SECJ /SECP/SECR/SECV)
2 (SCSR/SCSV)
TIME : 2.45 pm – 4.45 pm (2 Hours)
DATE/ DAY : 30th APRIL 2020 (THURSDAY)

INSTRUCTIONS TO THE STUDENTS:

- Please read the ***General Guidelines for the Programming Technique II Test*** that is shared in WhatsApp's Group and/ or provided in UTM e-learning
- Read the problem and instructions carefully.
- You are given **TWO HOURS** to complete the test inclusive the submission of your program.

IMPORTANT NOTES:

- All the **COMMENT STATEMENTS** in the submitted program **WILL NOT BE EVALUATED**.

SUBMISSION PROCEDURE:

- Only the source code is required for the submission and the source code's file shall be named as follows: ***Name_ matricsNo_section.cpp*** (i.e. *AinaAli_A19EC018_01.cpp*).
- You do not need to compress the file.
- Submit the source code file (i.e., sbt1.cpp) via the **UTM's e-learning system**.

Problem [100 Marks]

Write a complete C++ program that contains a class called **Person**.

1. The class should have six (6) **`private`** member variables (attributes) to hold a name, National Registration Identity Card (NRIC) number that has 12-digit numbers, height in meter, weight in kilogram, Body Mass Index (BMI) and weight status of BMI.
   Note: You must create the object of **`string`** class to store/ hold the NRIC number.

2. The class should have the following **`public`** member functions:

   - A **constructor with default arguments** that assign the values of name, height and weight member variables with the passed arguments. The constructor also enables to assign empty string ("") to the name and NRIC member variables and zero (0.0) to the height and weight member variables. It is also need to invoke **calcBMI** function to calculate the value of BMI.
   - A **getName** function that returns the value of name member variable.
   - A **getStatus** function that returns the value of weight status of BMI member variable. The weight status of BMI is based on the following table:

BMI	Weight Status
Below 18.5	Underweight
18.5 ≤ BMI < 25	Normal
25 ≤ BMI < 30	Overweight
30 and above	Obese

   - A **getRisk** function that returns the health risk based on the following table:

BMI	Health Risk
Below 18.5	Risk of nutritional deficiency diseases and osteoporosis
18.5 ≤ BMI < 23	Low risk
23 ≤ BMI < 27.5	Moderate risk
27.5 and above	High risk

   - A **calcBMI** function that calculates the value of BMI member variable according to the following formula: **BMI = weight / height^2**, where weight is in kilogram (kg) and height in meter (m).
   - A **readInput** function that reads the values from the keyboard for name, NRIC, height and weight member variables. It is also need to invoke **calcBMI** function to obtain the value of BMI.

- An **overloaded equality (==) operator** that determines if two objects have the same weight status, then return **true**; otherwise return **false**.

3. The class should also have the following **`friend`** of standalone functions:
   - A **dispInfo** function to display the age and gender of Person's object in the following format (please refer to the sample output given in **Figure 1)**:

     ```
     You are ## year old ?????
     ```

     The age will be determined by the first two digits of NRIC number, whereas the gender will be determined by the last digit of NRIC number. If the last digit is odd number, then the Person's object is man; otherwise the Person's object is woman.

     Example: *NRIC number is 980203041527. The first two digits are 98, meaning the Person's object was born in 1998. His/ her age is 22. The last digit is 7. It is an odd number. Therefore, the Person's object is man.*

   Note: You must use an appropriate function(s) in **`string`** class to determine the age and gender of Person's object.

   - An **overloaded output (<<) operator** function to display the object in the following format (please refer to the sample output given in **Figure 1**):

     ```
     Hello, ????? ?????
     You are ## year old ?????
     Your height is #.## meter
     Your weight is #.## kg
     Your BMI is #.##
     Your weight status is ?????
     You have ?????
     ```

4. Write an appropriate **main** function that do the following tasks:
   - Create two **Person** objects named **p1** and **p2**. The objects have the following data:

Object	p1	p2
Name	""	Anis Hashim
Height	0.0	1.5
Weight	0.0	50

   - Using an appropriate function(s), read the required input values and print the data of the appropriate object along with the notes about weight status for two objects (**p1** and **p2**) either both have same or different weight status.

Figure 1 shows a sample screen output that your program should produce. Note that, **bold** texts indicate keyboard input entered by the user. **Table 1** is the assessment criteria for the program.

```
BMI Calculator

<<< Input >>>
Name: Amir Arshad
NRIC Number (without hyphen'-'): 001221035161
Height (in meter): 1.82
Weight (in kg): 75

<<< Output >>>
Hello, Amir Arshad
You are 20 year old man
Your height is 1.82 meter
Your weight is 75.00 kg
Your BMI is 22.64
Your weight status is normal
You have low health risk

**Note: Amir Arshad and Anis Hashim have same weight status.
```

(a) Example of run 1

```
BMI Calculator

<<< Input >>>
Name: Safiah Rashid
NRIC Number (without hyphen'-'): 850624016242
Height (in meter): 1.54
Weight (in kg): 41.5

<<< Output >>>
Hello, Safiah Rashid
You are 35 year old woman
Your height is 1.54 meter
Your weight is 41.50 kg
Your BMI is 17.50
Your weight status is underweight
You have a risk of nutritional deficiency diseases and osteoporosis

**Note: Safiah Rashid and Anis Hashim have different weight status.
```

(b) Example of run 2

Figure 1: Example runs of program

Table 1: Assessment criteria

Item	Criteria	Marks
A	i) The program is able to run.	5
	ii) Using an appropriate structure for the program - the code is properly indented - all the required header files are included - the main function is properly written including proper input and output formatting used	 1 1 2
B	The definition of class Person i) Data members ii) Constructor iii) getName function iv) getStatus function v) getRisk function vi) calcBMI function vii) readInput function viii) Overloaded operator==	1 4 8 2 9 9 3 7 7
C	The definition of friend functions i) dispInfo function ii) Overloaded output (<<) operator	 19 11
D	Function main i) Object p1 and p2 creation ii) Read the required input values iii) Display the data of appropriate object iv) Display the notes about weight status for two objects	 3 1 1 6
Total		**100**