CSCI1530 Computer Principles and Java Programming 2018-2019 Second Term
Department of Computer Science and Engineering
The Chinese University of Hong Kong

Due date: 13 February 2019 (Wed) **Assignment 2** Full mark: 100
Expected normal time spent: 5 hours

Aim: 1. use of class **`Scanner`** to obtain keyboard input from console/ terminal.
2. use of class **`JOptionPane`** to create dialogues (Graphical User Interface/ GUI.)
3. practise declaring variables and performing calculations.

Exercise Zero: Background Knowledge:

1. Use the Java Standard API Class **`JOptionPane`** to create dialogues in a Java Application. Here is an example **`SampleDialogue.java`** for your reference:

```
import java.util.Scanner;
import javax.swing.JOptionPane;

class SampleDialogue
{
    public static void main(String[] args)
    {
        String answer;
        answer = JOptionPane.showInputDialog("Input PI");
        // parse/ convert the answer (which is a String) to a number (a double)
        double number;
        number = Double.parseDouble(answer);

        // re-use variable answer to store another piece of user input text
        answer = JOptionPane.showInputDialog("Input your SID");
        // parse/ convert the answer (which is a String) to a SID (an integer)
        int SID;
        SID = Integer.parseInt(answer);

        JOptionPane.showMessageDialog( null,
               "Hi student " + SID + ", PI is " + number + ", isn't it?");

        // connect to the keyboard using class Scanner and System.in
        Scanner keyboard = new Scanner(System.in);

        // ask questions and get inputs via the console/ terminal
        System.out.print("Type a line and <Enter>: ");
        String aLineOfText = keyboard.nextLine();
        System.out.print("Type a number and <Enter>: ");
        int anIntegerNumber = keyboard.nextInt();

        System.out.println("Got " + anIntegerNumber + " after " + aLineOfText);
        // program terminates here after the last user interaction
    }
}
```

2. Explanations: both **`JOptionPane`**, **`Integer`** and **`Double`** are classes provided in the Java Application Programming Interface **應用程式介面** (API). Without discussing much about them, first of all, let's try using them! They provide us some methods (actions!) that we can make use of. We are going to *send messages* in order to achieve our goal of implementing the system.

3. Class **`Scanner`** used in conjunction with **`System.in`** helps us getting text and number inputs from the console.

Copyright © 2019 CUHK, created by Michael Ping-Fu, FUNG

Tasks:

1. Create a N E W NetBeans Project **JHealth** with a package named **jhealth** and a new Java class **JHealth**, i.e., **Create Main class jhealth.JHealth**. Type your name, student ID and declaration statement clearly at the top comment block of the file **JHealth.java**.

```
/**
 * CSCI1530 Assignment 2 JHealth
 * Aim: user interaction and simple calculations
 *
 * Declaration: . . . <please refer to Assignment 1 for our requirements>
 * Student Name: xxx <fill in yourself>
 * Student ID  : xxx <fill in yourself>
 */
```

4. In the method **main()**, ask the user for some basic information using **Scanner** and **System.in**, a sample *console* interaction follows:

```
Texts in blue are user inputs followed by <Enter>, pay attention to the spaces
12345678901234567890123456789012345678901234567890123456789012345678901234567890 Ruler for reference
Name: Peter PAN
Choose an app below
1) HeartRate
2) BMI
Choice: 1
Showing you an input dialog...
Then show the corresponding app dialogue to continue interacting with the user…
12345678901234567890123456789012345678901234567890123456789012345678901234567890 Ruler for reference
```

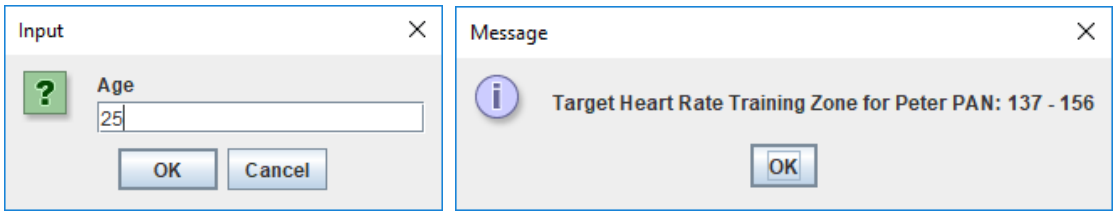
5. If the choice is 1, the program shall ask for **Age** (an integer) of the user using class methods **JOptionPane.showInputDialog()** and **Integer.parseInt()**. It then calculates the Heart Rate Training Zone of the user using the following formula:

   a) Maximum heart rate (beat per minute) = 220 – age; where 220 is a magic number/ constant

   b) Target heart rate zone = 70-80% of the Maximum heart rate; again the range is fixed

   c) Lower target heart rate = **Math.round**(Maximum heart rate x 0.7)
   d) Upper target heart rate = **Math.round**(Maximum heart rate x 0.8)
   e) You may need to resolve data type issues in the calculation and rounding conversion.

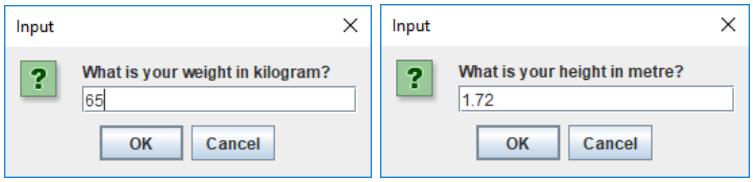
   f) The program then displays the calculated target heart rate training zone using **JOptionPane.showMessageDialog()**. It terminates after the user clicking on the Ok button on the Message dialog box.



Copyright © 2019 CUHK, created by Michael Ping-Fu, FUNG

g) Notice that the input dialog box may be shown behind the NetBeans window. Try switching between tasks and windows on your desktop.

6. If the choice is 2, the program shall ask for **weight** (in kg) and **height** (in m) of a user using input dialog boxes one-by-one. These two double-type input values may contain decimal places.



7. Because the user types some "text" (String) in respond to the weight and height questions, we have to "parse" (i.e., analyze) the textual inputs and convert the "text" to numbers using the example code demonstrated in the **`SampleDialogue`** example. In case the conversion fails, our Java program will be *terminated exceptionally*. This is an expected and accepted behaviour in this exercise.

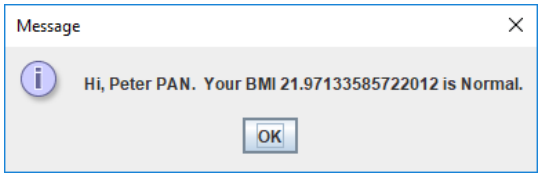
8. It then calculates the Body Mass Index (BMI) of the user using the following formula:

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

where weight is in kilogram, height is in metre. We ignore –ve and division-by-zero scenario.

9. The program displays the result in another Message dialog box with default number of decimal places and a diagnosis based on Asian Adult. It terminates after the user clicking on the Ok button on the Message dialog box.

Diagnosis	Erroneous	Low	Normal	Pre-obese	Obese
BMI	< 5	≥ 5 and < 18.5	≥ 18.5 and < 23.0	≥ 23.0 and < 27.5	≥ 27.5



10. We **need not validate** (check) the input of the user. The user is assumed to be rational and cooperative.

There is NO sample program for this exercise. Take the above sample usage sessions as a guide for input/ output screen formatting.

Running and Submission:

1. Since there are two Main classes in the same project, we shall "Run File (Shift-F6)" to test run part A and part B respectively. Run Project will always run the default Main Class defined in the Project Properties → Run.

Copyright © 2019 CUHK, created by Michael Ping-Fu, FUNG

2. Zip and Submit your *whole NetBeans project folder* in an archive file **`JHealth.zip`** (ZIP instead of RAR or other file formats) via our Online Assignment Collection Box on Blackboard <`https://blackboard.cuhk.edu.hk`>. After uploading the file, be reminded to click the **Submit** button to complete the submission.

Marking Scheme and Notes:

1. The submitted program should be free of any typing mistakes, compilation errors and warnings.

2. Comment/remark, indentation, style are under assessment in every programming assignments unless specified otherwise. The sample program gives you an example of a well-formatted source file. Variable naming, proper indentation for code blocks and adequate comments are important.

3. We will check and grade your work on correctness of your submission file type and filename, NetBeans project filename, package name, Java class filename, accuracy of your SID and the name in the declaration comment block, functionalities, input/ output message correctness, etc.

4. Remember to do your submission by the due date. No late submission would be accepted.

5. If you submit multiple times, **ONLY** the content and time-stamp of the **latest** one would be counted. You may delete (i.e. take back) your attached file and re-submit. We ONLY take into account the last submission.

University Guideline for Plagiarism

Attention is drawn to University policy and regulations on honesty in academic work, and to the disciplinary guidelines and procedures applicable to breaches of such policy and regulations. Details may be found at http://www.cuhk.edu.hk/policy/academichonesty/. With each assignment, students will be required to submit a statement that they are aware of these policies, regulations, guidelines and procedures.

Copyright © 2019 CUHK, created by Michael Ping-Fu, FUNG