The University of Hong Kong
Faculty of Engineering
Department of Computer Science

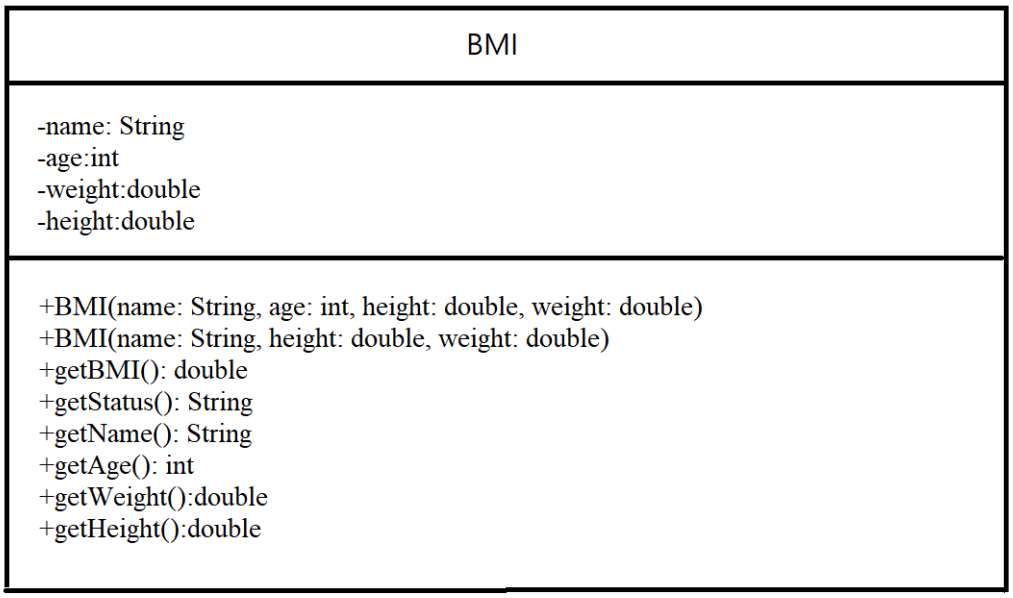
COMP2396B Object-oriented Programming and Java
Mid-term Examination

Date: 26th March, 2021 Time allowed: 1 hour 30 minutes

Note: Please read carefully!

1. There are 2 questions in this mid-term examination. Please complete both questions and submit your work using VPLs in Moodle.
2. All code on this question paper can be found in the text file *2396mt.txt* which can be downloaded from Moodle.
3. Don't communicate with your classmates during the mid-term examination in any means.
4. Double check that you have submitted all your files before leaving.
5. Static variables are evil - a static variable will keep data across objects and test cases. Think carefully before using it.

Question 1:



Write a program in Java to implement the above BMI class. Here are the requirements on the data field/constructors/methods for the BMI class:

Note: + means public while - means private.

- name (i.e. the name of the person) does not have any default value.
- age (i.e. the age of the person) does not have any default value.
- weight (i.e. the weight of the person in pounds) does not have any default value.
- height (i.e. the height of the person in inches) does not have any default value.
- The constructor that is with the argument **name, age, height and weight** should create a BMI object with the specified name, age, height and weight.
- The constructor that is with the argument **name, height and weight** should create a BMI object with the specified name, a default age 25, height and weight.
- The method **getBMI** should return the BMI value of the person (round up to 2 decimal places). Please note that the BMI is calculated as (weight in kilograms) / (height in meters)^2. Please note that one pound is equal to 0.45359237 kilograms while 1 inch is equal to 0.0254 meters.
- The method **getStatus** should return the BMI status of the person. The status is defined as "seriously underweight" if the calculated BMI <16; "underweight" if the calculated BMI < 18; "normal weight" if the calculated BMI <24; "over weight" if the calculated BMI < 29; "seriously over weight" if the calculated BMI < 35 and "gravely over weight" if the calculated BMI > 35.
- The method **getName** should return the name of the person.
- The method **getAge** should return the age of the person.
- The method **getWeight** should return the weight of the person.

- The method **getHeight** should return the height of the person.

Sample main()	`public class UseBMIClass {`
	`  public static void main (String []args) {`
	`  BMI b1= new BMI ("Ken", 18, 72, 150);`
	`  System.out.println("The BMI for " + b1.getName() + " is " +`
	`  + b1.getBMI() + " "+ b1.getStatus());`
	`  BMI b2= new BMI ("Dan", 70, 240);`
	`  System.out.println("The BMI for " + b2.getName() + " is " +`
	`  + b2.getBMI() + " "+ b2.getStatus());`
	`  }`
	`}`

Sample output	`The BMI for Ken is 20.34 normal weight` `The BMI for Dan is 34.44 seriously over weight`

Submit your BMI.java to Moodle.

Question 2:

Heung Shing Exhibition Center uses an event logging system for keeping tracks of guest's activities in events. Events are somethings too big and should be held in different venues. **Note that a guest can join multiple events and can join the same event in another venue**. Due to an epidemic, the system can check whether pairs of guests have joined the same venue of the same event to determine whether they have close contact.

You are given a sample main() below. Creates all classes to complete the program. Note that hasCloseContactWith() should return a boolean type value. Hint: Use ArrayList to help with your implementation.

```
public class Main {

    public static void main(String[] args) {
        Guest wing = new Guest("Wing");
        Guest joy = new Guest("Joy");
        Guest marco = new Guest("Marco");

        Event bookFair = new Event("Book Fair");
        bookFair.addVenue("Hall A");
        bookFair.addVenue("Room 1");

        wing.joinEvent(bookFair, "Hall A");
        joy.joinEvent(bookFair, "Room 1");
        marco.joinEvent(bookFair, "Room 1");

        Event gunplaExpo = new Event("Gunpla Expo");
        gunplaExpo.addVenue("Room 1");

        wing.joinEvent(gunplaExpo, "Room 1");

        System.out.println(wing.hasCloseContactWith(joy)); //output: false
        System.out.println(wing.hasCloseContactWith(marco)); //output: false
        System.out.println(joy.hasCloseContactWith(marco)); //output: true
    }

}
```

Output:

```
false
false
true
```

Submit your classes, **excluding** the Main.java, to Moodle. You can upload up to 5 files for this question.

END OF MID-TERM EXAMINATION