CSC 1109 Lab 4

Woon Jun Wei, 2200624 January 18, 2023

Question 1

Method assignments

I first instantiated 3 variables: ‘heightInInches‘, ‘weightInPounds‘ and ‘bmi‘. The constructor takes in 2 parameters, ‘heightInInches‘ and ‘weightInPounds‘ and assigns them to the corresponding variables.

The getters and setters are required to access the variables from outside the class, which was required for the ‘main‘ method in BMI_Main.java.

calBMI()

The ‘calBMI‘ method takes in 2 parameters, ‘heightInInches‘ and ‘weightInPounds‘. Using these variables, I first converted the **height** from inches to meters, using the formula ‘heightInMeters = heightInInches * 0.0254‘ where 0.0254 is given as the conversion ratio in the lab handout and I converted the **weight** from pounds to kilograms, using the formula ‘weightInKilograms = weightInPounds * 0.45359237‘ where 0.45359237 is given as the conversion ratio in the lab handout.

I then calculated the **BMI** using the formula ‘bmi = weightInKilograms / (heightInMeters * heightInMeters)‘.

BMIInterpretation()

The ‘BMIInterpretation‘ method takes in 1 parameter, ‘bmi‘ and returns a string based on the value of ‘bmi‘.
The method checks against values in the table provided in the lab sheet and returns the corresponding string.

Listing 1: BMI.java

```java
public class BMI {
    private double heightInInches;
    private double weightInPounds;
    private double bmi;

    public BMI(double heightInInches, double weightInPounds){
        this.heightInInches = heightInInches;
        this.weightInPounds = weightInPounds;
    }

    public double getHeightInInches() {
        return this.heightInInches;
    }

    public double getWeightInPounds() {
        return this.weightInPounds;
    }
    public void setHeightInInches(double heightInInches) {
        this.heightInInches = heightInInches;
    }

    public void setWeightInPounds(double weightInPounds) {
        this.weightInPounds = weightInPounds;
    }

    public double calBMI(double heightInInches, double weightInPounds){
        double heightInMeters = heightInInches * 0.0254;
        double weightInKilograms = weightInPounds * 0.45359237;
        this.bmi = weightInKilograms / (heightInMeters * heightInMeters);
        return this.bmi;
    }

    public void BMIInterpretation() {
        if (this.bmi < 18.5) {
            System.out.println("Underweight");
        } else if (this.bmi >= 18.5 && this.bmi < 25) {
            System.out.println("Normal");
        } else if (this.bmi >= 25 && this.bmi < 30) {
            System.out.println("Overweight");
        } else if (this.bmi >= 30) {
            System.out.println("Obese");
        }
    }
}
```

BMI_Main.java

The ‘BMI_Main‘ class uses the ‘BMI‘ class to calculate the BMI of a person. This is the main function that was used to access an use the ‘BMI‘ class.
Using a Scanner, I first got the hight and weight from the user, I then created a new ‘BMI‘ object and passed the height and weight to the constructor. I called the ‘calBMI‘ method and passed the height and weight to the method.

Lastly, the ‘BMIInterpretation‘ method was called to print the BMI interpretation. The input and output variables from the first output corresponds with the output from the lab sheet

Listing 2: BMI_Main.java

```java
import java.util.Scanner;

public class BMI_Main {
    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);

        System.out.print("Enter your weight in pounds: ");
        double weightInPounds = s.nextFloat();

        System.out.print("Enter your height in inches: ");
        double heightInInches = s.nextFloat();

        BMI bmi = new BMI(heightInInches, weightInPounds);
        System.out.println("BMI is " + bmi.calBMI(bmi.getHeightInInches(),↩
            bmi.getWeightInPounds()));

        bmi.BMIInterpretation();
    }
}
```

Listing 3: BMI_Main.java Output

```
//Output 1
Enter your weight in pounds: 146
Enter your height in inches: 70
BMI is 20.948603801493316
Normal

//Output 2
Enter your weight in pounds: 150
Enter your height in inches: 80
```

```
BMI is 16.478193272792794
Underweight
```

Question 2

Listing 4: StackOfIntegers.java

```java

public class StackOfIntegers {
    private int[] elements;
    private int size;

    public StackOfIntegers(){
        this.elements = new int[16];
        this.size = -1;
    }
    public StackOfIntegers(int capacity){
        this.elements = new int[capacity];
        this.size = -1;
    }

    public boolean empty(){
        return size == -1;
    }
    public int peek(){
        return elements[size - 1];
    }

    public void push(int value){
        this.size++;
        this.elements[size] = value;
    }
    public int pop(){
        int value = elements[size];
        this.size--;
        return value;
    }
    public int getSize(){
        return size;
    }
}
```

Listing 5: TestStackOfIntegers.java

```java
public class TestStackOfIntegers {
    public static void main(String[] args) {
        StackOfIntegers stack = new StackOfIntegers();
        for (int i = 0; i < 10; i++) {
            stack.push(i);
        }
        while (!stack.empty()) {
            System.out.print(stack.pop() + " ");
        }
    }
}
```

Listing 6: TestStackOfIntegers.java Output

```
9 8 7 6 5 4 3 2 1 0
```