**CS 250 - Fall 2021 - HW#2**

Due: Sep 15, 2021, 11:59 pm

# Objectives

The objectives of this assignment are:

* Familiarizing with the problem-solving steps: analysis, design, implementation, and testing.
* Practice editing, compiling, running, and revising a Java application.
* Practicing exception handling
* Practicing using interface
* Practicing I/O programming

# Part 1. Non-programming Questions ( 50 points)

**Q1. (18 points) Choose one correct answer for each sub-question below.**

1. **Which type of exception that Java requires to be explicitly handled or declared in the method it may be thrown?**

|  |  |
| --- | --- |
| a. Checked exception | b. Unchecked exception |
| c. Both checked and unchecked | d. None of them |
|  |  |

1. **A subclass can only inherit \_\_\_\_\_\_\_\_\_ attributes and methods from its superclass.**

a. public b. protected

c. public and protected d. all except protected

**3. In a Java program, we can throw an exception inside the \_\_\_\_\_\_\_\_\_ .**

a. try block b. catch block

c. main method d. finally block

**4. The root of Java’s exception class hierarchy is the \_\_\_\_\_\_\_\_\_ class.**

a. Exception b. Throwable

c. IOException d. RuntimeException

**5. In Java, \_\_\_\_\_\_\_\_\_\_ files are more portable and platform independent.**

a. binary b. text

c. both text and binary d. raw

**6. In Java, a class can extend \_\_\_\_\_\_\_\_ superclass (parent class) and implement \_\_\_\_\_\_ interface.**

|  |  |
| --- | --- |
| a. only one, only one | b. only one, more than one |
| c. more than one, only one | d. more than one, more than one |

# Q2. (16 points)

Implement the following method (complete Java code below the //TODO comment:

|  |
| --- |
| /\*\*   * This method receives the height and weight of a person, calculate BMI,\* and classify the body mass * @param weight: weight in pounds (lb) * @param height: height in inches (ft) * @return: classification string: Underweight, Normal, Overweight, and Obese   \*/  public static String calcBMI(double weight, double height) {  //TODO add code below  double bmi = 0;  String state = null;  if (weight > 0 && height > 0) {  bmi = weight / (Math.*pow*(height,2)) \* 703;    if (bmi < 18.5) state = "Underweight";  else if (bmi >= 18.5 && bmi < 25) state = "Normal";  else if (bmi >= 25 && bmi < 30) state = "Overweight";  else state = "Obese";  } else { throw new IllegalArgumentException("Invalid inputs") ;    }    return state;    } |

Below is the requirements and guidelines for this method:

* You need to check if the weight and height parameters are valid (should be positive). If they are invalid, throw a new instance of IllegalArgumentException with the error message “Invalid inputs!!!”
* If two parameters are valid, calculate the BMI using the equation: BMI = weight

(lb) / [height (in)]2 x 703

* Using if-else or switch on the value of BMI and return the classification as

|  |  |
| --- | --- |
| follows: |  |
| - BMI < 18.5 | => Underweight |
| - 18.5 <= BMI < 25 | => Normal |
| - 25.0 <= BMI < 30 | => Oveweight |
| - BMI >= 30 | => Obese |

# Q3. (16 pts)

Analyze the following code

|  |
| --- |
| public static void hw2\_q3(int num) { try {  System.out.println( "Begin the try block code" ) ; |
| if (num > 100) throw new Exception (num + " is too large" ) ; System.out.println( "End the try block code" ) ;  } catch ( Exception e ) {  System.out.println( "ERROR: " + e.getMessage ( ) ) ;  }  } |

Determine the output of the Java program if you call such above method in the main() method with the following inputs and justify your answers (explain why such outputs happen):

1. hw2\_q3(0)

**It will print everything except the exception, since there is no error:**

**Begin the try block code**

**End the try block code**

1. hw2\_q3(101)

**It will print the first system.out.print & the exception message, since 101 is greater than 100 and will cause the error and avoid printing the last part:**

**Begin the try block code**

**ERROR: 101 is too large**

1. hw2\_q3(150.0)

**It will print the first system.out.print & the exception message, since 150 is greater than 100 and will cause the error and avoid printing the last part:**

**Begin the try block code**

**ERROR: 150 is too large**