Object Oriented Programming (JAVA)

23W CST 8284

Assignment 1

Instructions

This assignment is designed to be completed and submitted by you **individually**. It is your responsibility to ensure that your work is NOT **replicated** by any other student, and that you do not copy or submit someone else's work. Avoid plagiarism.

The **due date** for submission of this assignment is **on Brightspace. The final date to demo this assignment is in Week #9 (whichever day your ab session that day falls into).**

C**arefully** read every instruction and the tasks to ensure that you deliver all aspects of your solution as required. This assignment when completed fully and correctly **weighs 8%** of the entire course marks. Ask for clarification if needed.

Problem Description

Electronic Health Records (EHR) is one of the most innovative items in recent healthcare systems. It enables the digitization of health records of patients to be shared easily using computer systems among care providers to facilitate better patient care.

In every patient's electronic record, certain information must be presented. In this assignment, your task will be to facilitate the creation of patients' health records during their first time of hospital visit.

Ensure that you completely review the **code file** and the **UML Class** diagram **provided for you** in order to **understand** what they contain and the **modification** you are required to make.

COMPONENT 1

1. **Download, review** and **modify** the java class file provided for you named Patient.java.

2. Take time to study the **UML class diagram** that has been provided for you, to see all the **attributes, methods, etc.**, which are required to help guide your solution.

3. The **attributes** of Patient.java should include the following patient's data:
   a. First name
   b. Last name
   c. Gender
   d. Height (in inches)
   e. Weight (in pounds)
   f. **Hint 1:** Check the **UML class diagram** provided for you and ensure that you **include** all other **attributes** that have been listed in the class.

4. The class (Patient) should have a constructor that receives these data as parameters.

5. For each attribute mentioned in (2) above, you should provide the appropriate ***set*** and ***get*** methods.

6. The class must provide ***methods*** to calculate and return the patient's:
   a. Age (in years)
   b. *MaximumHeartRate* (**Hint 2:** *MaximumHeartRate* is calculated as: ***220 – Age in years)***
   c. Target heart rate range (The Hints to calculate the two components in this range are provided below):
   Hint 3: *MinimumTargetHeartRate* is calculated as: 50% of *MaximumHeartRate*.

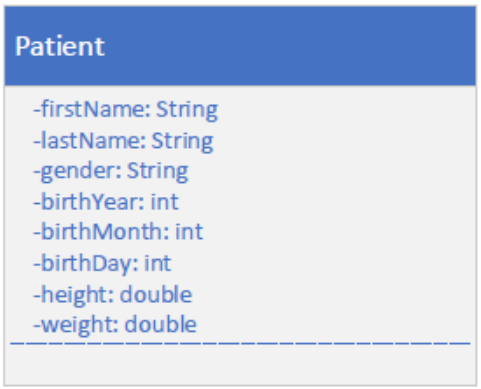
Hint 4: *MaximumTargetHeartRate* is calculated as: 85% of *MaximumHeartRate*.

d. Body Mass Index (BMI) (**Hint 5:** This calculation has already been provided for you)

7. Write a Test class for your code named MyHealthDataTest.java. This code must **prompt for input** of the patient's data (described above), **instantiate an object** of the class Patient for that patient and then prints the data from that object.
   The Health data to be **displayed** must include the patient's:
   a. First name
   b. Last name
   c. Gender
   d. Year of birth, month of birth, day of birth
   e. Height
   f. Weight
   g. Calculate and print patient's age in years, BMI, maximum heart rate, Target heart rate range (*MinimumHeartRate* and *MaximumHeartRate*).
   h. It should also display the **BMI values** (check your code to see this value).
8. Execute your code and include your **output file.** Your output file must be **eclipse screen shot only**. Any other form of representation of your output in this assignment will NOT be accepted. Be sure to capture the portion of the eclipse screen that includes your output the title portion above the output that shows the time of the execution.

9. Use the **Javadoc style comments** to explain your work. You are required to generate and submit the Javadoc files.

10. **Submit your Java files** (containing your java code files) along with two screen capture for the test method results to your Assignment 1 folder in Brightspace. Submit these as separate files. While naming your files for submission, use the naming format (use this format to avoid losing marks): **Lab_Section_XXX_YourFirstName_YourLastName**.



UML class diagram.

COMPONENT 2

1. Create a **two test** cases for Patient in a class and name it MyHealthDataTest2 to test the getBMI() method. Write 2 test cases using different inputs/outputs from the getBMI() method.
2. In your first **test**, show a successful execution of your test with no errors. The test method should show a message that the test has passed successfully.
3. In your second **test** , show a failed execution of your test (as a result of introducing an error into your code) – The test method should show a message that the test has failed.
4. Using Javadoc style comments state any modification or assumption you have made and where you made it. These assumptions and/or modifications must be stated at the beginning of your **Test** file (MyHealthDataTest2).

Assignment Rubrics (8%)

- File submission – use of specified format and portal provided
- Attributes: include correct and complete declaration
- Constructor: specify constructor(s) within your solution as required in Patient
- **set** and **get** methods: include correct and complete setters and getters as required
- Calculate and return correctly: formulate and apply calculations as listed in items 6a – 6c above.
- Complete and correct implementation of all aspects of your solution as required (user input, instantiate an object, printing required data from the object, etc.
- Include Javadoc comments in the right places within the code. Correctly generate and demo the Javadoc files for Component 1
- Two test methods (with assumptions made)
- Eclipse output showing the successful execution of the test
- Eclipse output showing the failed execution of the test