

**CEGEP VANIER COLLEGE**  
**Department of Computer Science**

**Assignment-01: Object Oriented Programming in C#**

**Assignment-01 weight is 3% of final grade**  
**Max Marks (25)**

**Course Title** : Application Development-1 (Sec-03)  
**Course Number** : 420-331-VA

**Teacher** : Syed Naseem Afzal  
**Due Date** : September 09, 2025

**Problem-01: Target-Heart-Rate Calculator** **Max Marks (10)**

While exercising, you can use a heart-rate monitor to see that your heart rate stays within a safe range suggested by your trainers and doctors. According to the American Heart Association (AHA)

[www.heart.org/HEARTORG/GettingHealthy/PhysicalActivity/Target-Heart-Rates\\_UCM\\_434341\\_Article.jsp](http://www.heart.org/HEARTORG/GettingHealthy/PhysicalActivity/Target-Heart-Rates_UCM_434341_Article.jsp)

the formula for calculating your *maximum heart rate* in beats per minute is 220 minus your age in years. Your *target heart rate* is a range that is 50–85% of your maximum heart rate.

**[Note:** These formulas are estimates provided by the AHA. Maximum and target heart rates may vary based on the health, fitness and gender of the individual. Always consult a physician or qualified health care professional before beginning or modifying an exercise program.]

1. Create a class called **HeartRates**. The class attributes should include the person's
  - i First Name
  - ii Last Name
  - iii Birth Year
  - iv Current Year.
2. Your class should have a constructor that receives this data as parameters to initialize the instance variables.
3. For each attribute provide a property with **set** and **get** accessors.
4. The class also should include
  - i a **property** that calculates and returns the person's age (in years),
  - ii a **property** that calculates and returns the person's maximum heart rate and
  - iii **properties** that calculate and return the person's minimum and maximum target heart rates.
5. The class **HeartRates** should have a public method **DisplayPatientRecord**, that display patient's data in tabular form as shown below, using C# “**String Interpolation**” to insert values in string literals to create formatted strings, and format specifier to format numeric values and alignment.

<b>PATIENT HEART RATE RECORD</b>	
Patient Name	LastName, FirstName
Patient Birth Year	1971
Patient Age	50
Maximum Heart Rate	170
Targe Heart Rate Range	85—144

Write a driver class (app) that prompts for the person's data input, instantiates an object of class **HeartRates** and displays the patient's information from that object by calling the **DisplayPatientRecord**, method.

**[Note:** Please use “**String Interpolation**” to insert values in string literals to create formatted strings of your app output].

## Problem-02: Computerization of Health Records

Max Marks (15)

A health care issue that has been in the news lately is the computerization of health records. This possibility is being approached cautiously because of sensitive privacy and security concerns, among others. Computerizing health records could make it easier for patients to share their health profiles and histories among their various health care professionals. This could improve the quality of health care, help avoid drug conflicts and erroneous drug prescriptions, reduce costs and, in emergencies, could save lives.

1. Design a **HealthProfile** class for a person. The class attributes should include the person's
    - i First Name
    - ii Last Name
    - iii Birth Year
    - iv Height (in inches)
    - v Weight (in pounds)
    - vi Current Year.
  2. Your class should have a constructor that receives this data as parameters to initialize the instance variables.
  3. For each attribute provide a **property** with **set** and **get** accessors.
  4. The class also should include
    - a) a **property** that calculates and returns the person's age (in years),
    - b) a **property** that calculates and returns the person's maximum heart rate and
    - c) **properties** that calculate and return the person's minimum and maximum target heart rates.
    - d) a method that calculate and return the body mass index (BMI) using the following formula:
- $$BMI = \frac{weightInPounds \times 703}{heightInInches \times heightInInches}$$
- e) a method that return the BMI Text value depending on BMI computed numeric value
5. The class **HeartRates** should have a public method **DisplayHealthRecord**, that display patient's data in tabular form as shown below, using C# “String Interpolation” to insert values in string literals to create formatted strings, and format specifier to format numeric values and alignment.

PATIENT HEALTH RECORD	
Patient Name	LastName, FirstName
Patient Birth Year	1971
Patient Age	50
Patient Height	70
Patient Weight	200
Maximum Heart Rate	170
Targe Heart Rate Range	85—144
BMI Numeric Value	28.69
BMI Text Value	Overweight

Following table may be used to compute the “BMI Text Value” based on BMI computed numeric value.

BMI TEXT VALUES	
Less than 18.5	Underweight
Between 18.5 and 24.9	Normal
Between 25 and 29.9	Overweight
30 or greater	Obese

Write a driver class that prompts for the person's data input, instantiates an object of class **HealthProfile** and displays the patient's information from that object by calling the **DisplayHealthRecord**, method.

**[Note:** Please use “**String Interpolation**” to insert values in string literals to create formatted strings of your app output].

### What to hand in?

After successful demonstration, create a compressed file [**Filename: LastNameFirstNameAssignment01.ZIP**], with the following:

- a) MS-Word document [**Filename: LastNameFirstNameAssignment01.docx**] containing screen shots of your examples’ output window with input values and produced output/result.
- b) Your working application project folder(s).

Upload your **LastNameFirstNameAssignment01.ZIP** file in LEA’s assignment section.

**Note:** Your uploaded (ZIP) file should contain all files of the working examples’ project folders. Incomplete and / or non-working app’s files will not be graded and have zero value.

---

The following table showing the heart rate averages, so use them as a general guide.

Age	Target HR Zone 50-85%	Average Maximum Heart Rate, 100%
20 years	100-170 beats per minute (bpm)	200 bpm
30 years	95-162 bpm	190 bpm
35 years	93-157 bpm	185 bpm
40 years	90-153 bpm	180 bpm
45 years	88-149 bpm	175 bpm
50 years	85-145 bpm	170 bpm
55 years	83-140 bpm	165 bpm
60 years	80-136 bpm	160 bpm
65 years	78-132 bpm	155 bpm
70 years	75-128 bpm	150 bpm