

# CS101- Algorithms and Programming I

## Lab 07

---

**Lab Objectives:** Classes and Objects.

---

For all labs in CS 101, your solutions must conform to the CS101 style guidelines (rules!)

1. Create a class, `FitnessAssessment.java` that has the following data members and methods. All data members should be private.

**Data Members:**

- `birthdate`: user's birthdate stored as `LocalDate` object.
- `height`: height of user in meters.
- `weight`: weight of user in kilograms.
- `restingHeartRate`: beats per minute.
- `waist`: waist measurement in centimeters.
- `hips`: hip measurement in centimeters.
- `gender`: m for male, f for female.

**Methods:**

- Constructor: initialize all data members to values passed as parameters. Sets the birthdate using the `setBirthDate()` method.
- Getters and setters for all private attributes.
- `setBirthDate()`: takes a `String` date as a parameter and sets the birth date using the values passed as parameter. See the [LocalDate documentation/help](#) to create a `LocalDate`.
- `calculateTargetHeartRate()`: calculate and return the target heart rate. Target heart rate is 85% of the maximum heart rate.
- `calculateMaxHeartRate()`: calculate and returns the max heart rate. Found by subtracting the age from 220.
- `calculateWaistHipRatio()`: calculates and returns the waist measurement divided by the hip measurement.
- `calculateBMI()`: calculates and returns the body mass index. Formula:  $\text{weight} / \text{height}^2$
- `findBMICategory()`: Returns the `String` bmi category of the user. 30+ is Obese, [25 – 30) -> Overweight, [20 – 25) -> Normal, < 20 -> Underweight.
- `calculateAge()`: calculates and returns the age using user's birthdate and current system date (use `LocalDate birthDate` and `LocalDate current date`).
- `determineFitnessLevel()`: returns the fitness level according to the total number of positive fitness indicators. Users get 1 point for each of the following: BMI between 20 and 24 inclusive, resting heart rate below 70, waist/hip ratio for men below 1 or for women below 0.9. A count of 0, fitness level is POOR, 1: AVERAGE, 2: GOOD, 3: EXCELLENT.

2. Create a class, `FitnessApp.java` that does the following:

- Input the required user information.
- Create a `FitnessAssessment` object.
- Until the user terminates the application, display the menu shown, and perform the action selected by the user.

### Sample Run:

Enter birthdate (YYYY-MM-DD): 1999-06-02  
Enter gender (f)emale/(m)ale: f  
Enter height(m) weight(kg): 1.6 55  
Enter waist and hip measurments (cm): 32 40  
Enter resting heart rate: 75

-----MENU-----

- 1 - Calculate Target Heart Rate
- 2 - Calculate Waist/Hip Ratio
- 3 - Body Mass Index
- 4 - Display Fitness Level
- 5 - Quit

Enter choice:1

Target Heart Rate: 168.3

-----MENU-----

- 1 - Calculate Target Heart Rate
- 2 - Calculate Waist/Hip Ratio
- 3 - Body Mass Index
- 4 - Display Fitness Level
- 5 - Quit

Enter choice:2

Waist/Hip Ratio: 0.8

-----MENU-----

- 1 - Calculate Target Heart Rate
- 2 - Calculate Waist/Hip Ratio
- 3 - Body Mass Index
- 4 - Display Fitness Level
- 5 - Quit

Enter choice:3

Your BMI is: 21.5 Category: Normal

-----MENU-----

- 1 - Calculate Target Heart Rate
- 2 - Calculate Waist/Hip Ratio
- 3 - Body Mass Index
- 4 - Display Fitness Level
- 5 - Quit

Enter choice:4

Your fitness level is: GOOD

-----MENU-----

- 1 - Calculate Target Heart Rate
- 2 - Calculate Waist/Hip Ratio
- 3 - Body Mass Index
- 4 - Display Fitness Level
- 5 - Quit

Enter choice:5