|  |  |  |  |
| --- | --- | --- | --- |
| |  |  | | --- | --- | |  | ECS2301 Software Engineering and Project  Final project (100 marks) | |  |

## Instructions:

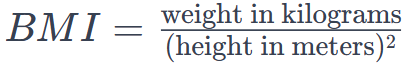
* This is a continuation of your mid-term exam project. You will complete the logic, input validation and output formatting here.
* If your SLTC student number ends with 0,1,2,3 then you must attempt question 1.
* If your SLTC student number ends with 4,5,6 then you must attempt question 2.
* If your SLTC student number ends with 7,8,9 then you must attempt question 3.
* Submit your answers as a single file (**.ZIP**) on or before the deadline provided in the LMS.
* Submission must include this document explaining your code, output screens and lessons learnt
* Late submission will not be considered for the marking.
* Make sure to include this document in your submission

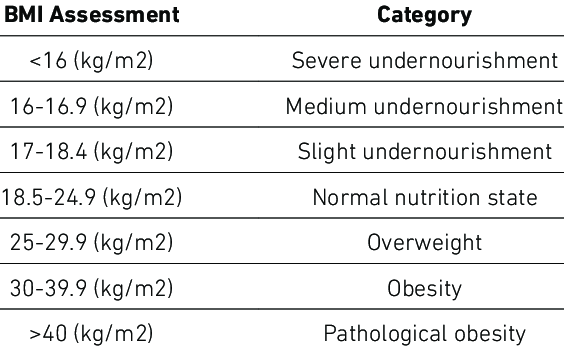
|  |  |
| --- | --- |
| Full name: | Mohamed Fareed Mohamed Asvi |
| Student index: | 22ug2-0100 |
| Date of submission: | 30th of march 2024 |

## Q1. BMI calculator

BMI, or Body Mass Index, is a numerical value of a person's weight in relation to their height. It is a commonly used screening tool to categorize individuals into different weight status categories, such as underweight, normal weight, overweight, and obesity.

BMI is calculated using the following formula:

Here's a breakdown of the BMI categories:

Watch this video for more details: https://youtu.be/t8sIioCX0lk

### Requirements:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Tester | | -id (int)  -name (String)  -yob (year of birth)  -height (int)  -weight (int) | | +main()  +displayMenu()  +index()  +view(int id)  +create()  +delete()  +exit() |   Tester class diagram | |  | | --- | | Bmi | | -id (int)  -name (String)  -yob (year of birth)  -height (int)  -weight (int) | | +bmi() (constructor)  +setters/getters for properties  +calculate()  +display() |   Bmi class diagram |

Write an application to store BMI of 5 users. It should be a menu driven application. You need to show the evidence of using classes, objects, methods, properties, setters/getters, constructors, abstraction, inheritance, polymorphism and java collections.

1. In your main class create a displayMenu() method and show the following choices. The program must continuously run until ‘exit’ is selected (10 marks) :
   1. Create a record. (Ask for user id, then ask data for name, year of birth, height & weight)
   2. Show BMI data for all users.
   3. Show BMI data for a selected user.
   4. Delete all.
   5. Exit application.
2. Write methods for all actions above, inside main class (10 marks for each)
   1. index() : to show all records. Call Bmi.display() here.
   2. view(int id) : to show one record for the given id. Include BMI category and recommendations if there are any. Call Bmi.display() here.
   3. create() : create a new record with appropriate input validation. Call Bmi constructor here.
   4. delete() : delete all records.
   5. exit() : exit to system
3. Use a suitable Java collection to store 5 users (10 marks)
4. Calculate BMI accurately, taking into consideration height in cm (5 marks)
5. Calculate the age with year of birth and current year (5 marks)
6. A section on the lessons learnt during this exercise (10 marks)
7. Generate JavaDoc files (5 marks)
8. Upload your complete, commented, tested code as a Git Hub repository to /final branch. Include the link in the document (5 marks)

# Paste the code here

GitHub repo url:

import java.util.ArrayList;

import java.util.Scanner;

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template

\*/

/\*\*

\*

\* @author Mohamed Asvi

\*/

public class Tester {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

ArrayList<Bmi> bmiRecords = new ArrayList<>();

Scanner scanner = new Scanner(System.in);

int choice;

do {

displayMenu();

System.out.print("Enter your choice: ");

choice = scanner.nextInt();

switch (choice) {

case 1:

for(int i = 0; i < 5 ; i++){

bmiRecords.add(createBmiRecord(scanner));

System.out.println("-------------------------");

}

break;

case 2:

for (Bmi bmi : bmiRecords) {

bmi.display();

System.out.println("----------------------");

}

break;

case 3:

int id = getId(scanner);

Bmi bmiToDisplay = getBmiById(bmiRecords, id);

if (bmiToDisplay != null) {

bmiToDisplay.display();

} else {

System.out.println("No BMI record found for the given ID.");

}

break;

case 4:

bmiRecords.clear();

System.out.println("All BMI records deleted successfully.");

break;

case 5:

System.out.println("Exiting the application.");

break;

default:

System.out.println("Invalid choice. Please choose again.");

}

} while (choice != 5);

}

private static Bmi createBmiRecord(Scanner scanner) {

System.out.print("Enter ID: ");

int id = scanner.nextInt();

System.out.print("Enter Name: ");

String name = scanner.next();

System.out.print("Enter Year of Birth: ");

int yob = scanner.nextInt();

System.out.print("Enter Height: ");

int height = scanner.nextInt();

System.out.print("Enter Weight: ");

int weight = scanner.nextInt();

return new Bmi(id, name, yob, height, weight);

}

private static int getId(Scanner scanner) {

System.out.print("Enter ID: ");

return scanner.nextInt();

}

private static Bmi getBmiById(ArrayList<Bmi> bmiRecords, int id) {

for (Bmi bmi : bmiRecords) {

if (bmi.getId() == id) {

return bmi;

}

}

return null;

}

private static void displayMenu() {

System.out.println("\nBMI Calculator Menu:");

System.out.println("1. Create a record.");

System.out.println("2. Show BMI data for all users.");

System.out.println("3. Show BMI data for a selected user.");

System.out.println("4. Delete all.");

System.out.println("5. Exit application.");

}

}

class Bmi {

private int id;

private String name;

private int yob;

private int height;

private int weight;

public Bmi(int id, String name, int yob, int height, int weight) {

this.id = id;

this.name = name;

this.yob = yob;

this.height = height;

this.weight = weight;

}

public int getId() {

return id;

}

public String getName() {

return name;

}

public int getYob() {

return yob;

}

public int getHeight() {

return height;

}

public int getWeight() {

return weight;

}

public double calculateBmi() {

return weight\*100\*100 / (height \* height);

}

public int Age(){

int age = 2024-yob;

return age;

}

public void display() {

System.out.println("\nID : " + id);

System.out.println("Name : " + name);

System.out.println("Year Of Birth : "+yob);

System.out.println("Heigth : "+height+"Cm");

System.out.println("Weigth : "+weight+"Kg");

System.out.println("Age : "+Age());

System.out.println("--------------------------");

System.out.println("BMI categories(Kg/m^2)\n------------------");

if(calculateBmi()<16){

System.out.println("Severe Undernourishment");

}

else if (calculateBmi()>=17 && calculateBmi() <= 18.4){

System.out.println("Medium Undernourishment");

}

else if (calculateBmi()>= 18.5 && calculateBmi() <= 24.9){

System.out.println("Sligth Undernourishment");

}

else if (calculateBmi()>=25 && calculateBmi() <=29.9){

System.out.println("Normal Nutrition State");

}

else if (calculateBmi()>=30 && calculateBmi() <=30.9){

System.out.println("Obesity");

}

else {

System.out.println("Pathological Obesity");

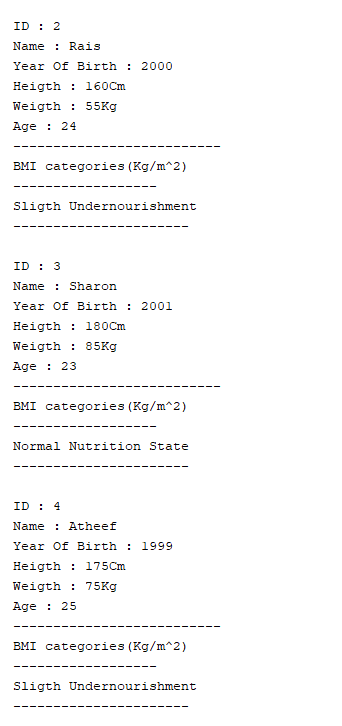
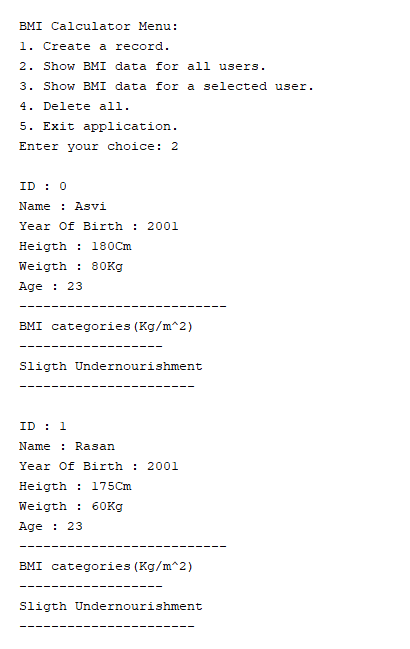
}

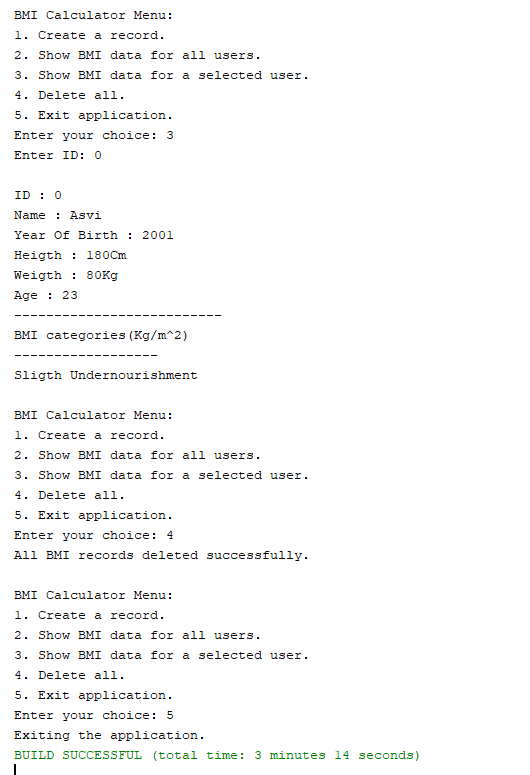
}

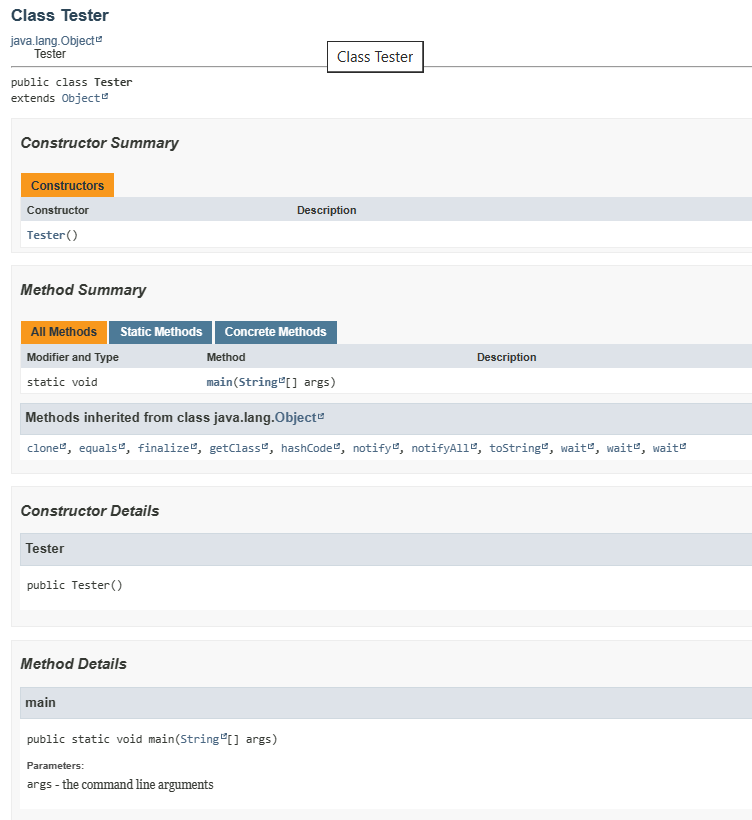
}

# Paste the output screens here







# Lessons learnt