# **DAYANANDA SAGAR UNIVERSITY**

Devarakaggalahalli, Harohalli Kanakapura Road, Dt, Ramanagara, Karnataka 562112



**Bachelor of Technology in**

**COMPUTER SCIENCE AND ENGINEERING**

**(Artificial Intelligence & Machine Learning)**

**SKILL ENHANCEMENT COURSE-I**

**(JAVA PROGRAMMING)**

**(23AM2306)**

**“BODY MASS INDEX”**

By

**Punya K Bhange(ENG23AM0059)**

**Sanjana BS(ENG23AM0068)**

**Under the supervision of**

**Prof. JEEVARAJ R**

**Assistant Professor,**

**CSE (AI & ML)**

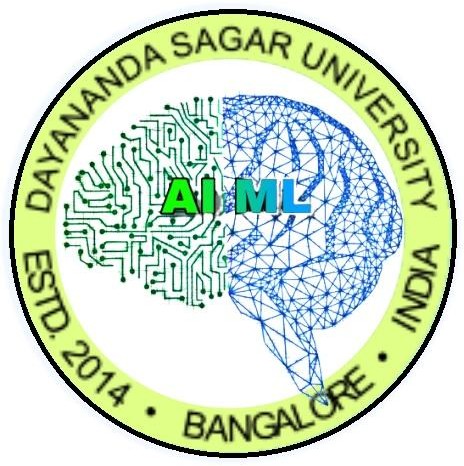
**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (AI&ML),**

#### SCHOOL OF ENGINEERING

#### DAYANANDA SAGAR UNIVERSITY, BANGALORE

**(2024-2025)**



**Day****ananda Sagar University**

**School of Engineering**

**Department of Computer Science & Engineering**

**(Artificial Intelligence & Machine Learning)**

**Devarakaggalahalli, Harohalli Kanakapura Road, Dt, Ramanagara, Karnataka 562112**

# CERTIFICATE

This is to certify that the **SKILL ENHANCEMENT COURSE-JAVA PROGRAMMING**

**(23AM2306)** work titled “**BODY MASS INDEX”** is carried out by

**PUNYA K BHANGE(END223AM0059),SANJANA BS(ENG23AM0068)** Bonafede students of Bachelor of Technology in Computer Science and Engineering (AI&ML) at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering( AI & ML), during the year **2024-2025**.

|  |  |
| --- | --- |
| --------------------------------  Prof. Jeevaraj R  Assistant Professor  Dept. of CSE (AI&ML),  School of Engineering  Dayananda Sagar University | -------------------------------------  Dr. Jayavrinda Vrindavanam V  Chairman CSE (AI&ML)  School of Engineering  Dayananda Sagar University |

##### 

ii

# DECLARATION

We, **Punya K Bhange and Sanjana BS** students of third semester B. Tech in **Computer Science and Engineering with speciation in Artificial intelligence and machine learning** , at School of Engineering, **Dayananda Sagar University**, hereby declare that the **SKILL ENHANCEMENT COURSE-JAVA PROGRAMMING (23AM2306)**titled **“Body Mass Index”** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2024-2025**.

Student 1: Signature

Name: PUNYA K BHANGE

USN: ENG23AM0059

Student 2: Signature

Name: SANJANA BS

USN: ENG23AM0068

**Place: Bangalore**

**Date:**

iii

**ACKNOWLEDGEMENT**

*It is a great pleasure for me to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.*

*First, I take this opportunity to express my sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor’s degree in this institution.*

*I would like to thank* ***Dr. Udaya Kumar Reddy K R, Dean, School of Engineering & Technology, Dayananda Sagar University*** *for his constant encouragement and expert advice.*

*It is a matter of immense pleasure to express our sincere thanks to* ***Dr. Jayvrinda Vrindavanam v, Department Chairman****,* ***Computer Science and Engineering(AI AND ML)****,* ***Dayananda Sagar University,*** *for providing right academic guidance that made the task possible.*

*I would like to thank our guide* ***Prof.Jeevaraj R, Assistant Professor,******Dept. of Computer Science and Engineering (AI AND ML)****,* ***Dayananda Sagar University****, for sparing his valuable time to extend help in every step of the project work, which paved the way for smooth progress and fruitful culmination of the project.*

*I am also grateful to my family and friends who provided me with every requirement throughout the course.*

*I would like to thank one and all who directly or indirectly helped me in the Project work*

iv

# TABLE OF CONTENTS

[LIST OF FIGURES ……….................................................................................... vi](#_TOC_250007)

ABSTRACT………………………………………………………………………. 01

[CHAPTER 1 INTRODUCTION………................................................................. 02](#_TOC_250005)

* 1. OBJECTIVE……………………………………………………………… 03

CHAPTER 2 PROBLEM DEFINITION. ………..................................................... 04

[CHAPTER 3 METHODOLOGY..................................................................](#_TOC_250002)..... 05-06

CHAPTER 4 IMPLEMENTATION OF CODE……………………………… 07-10

4.1 FULL CODE &OUTPUT……………………………………… ……. 11-15

CHAPTER 5 TECHNOLOGY USED ….............................................................. 16-17

CHAPTER 6 RESULT AND ANALYSIS………………………………………….. 18-19

CHAPTER 7 CONCLUSION……………………………………………………… 20

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| Fig. No. | Description of the figure | Page No. |
| 4(a) | Base class Representation | 07 |
| 4(b) | Variables | 07 |
| 4(c) | Methods | 08 |
| 4(d) | Logical Constructs | 08 |
| 4(e) | Input/Output Operations | 09 |
| 4(f) | Constants or Fixed Values | 09 |
| 4(g) | Output of the implemented code | 10 |

vi

# ABSTRACT

# Project focuses on the development of an Enhanced BMI Calculator, a comprehensive and user-friendly application designed to calculate Body Mass Index (BMI) and provide personalized health recommendations. This The system allows users to input their age, gender, and physical measurements using either the metric or imperial measurement systems. Based on the input, the program calculates BMI, categorizes it into standard health ranges (e.g., Underweight, Normal weight, Overweight, Obesity), and offers tailored health advice specific to the user's age and gender.

# The project highlights the effective use of object-oriented programming principles, including modularity, with distinct methods for BMI calculation, category determination, and recommendation generation. Key features of the system include dynamic input handling, conditional logic for category classification, and personalized guidance for health and fitness improvement. Special considerations, such as multiple input formats and age- and gender-specific advice, ensure accessibility and relevance to diverse user groups.

# This Enhanced BMI Calculator serves as a practical tool for promoting health awareness and assisting users in making informed lifestyle choices, showcasing a robust implementation of Java programming constructs like method overloading, input validation, and conditional branching. The system is designed to enhance user engagement and provide a reliable solution for BMI tracking and health management.

# CHAPTER 1

# INTRODUCTION

In today's health-conscious society, understanding and managing personal well-being has become a priority for individuals of all ages. Among the various health metrics, Body Mass Index (BMI) serves as a fundamental indicator for assessing body weight relative to height, providing insights into an individual's overall health status. Recognizing the importance of this metric, our project introduces the *Enhanced BMI Calculator*, an interactive and feature-rich application designed to simplify BMI calculation while offering personalized health guidance.

The primary goal of this project is to create a user-friendly tool that accommodates diverse user needs through the integration of both metric and imperial measurement systems. Users can input their age, gender, weight, and height to calculate their BMI, classify it into standard health categories (Underweight, Normal weight, Overweight, or Obesity), and receive tailored recommendations for improving or maintaining their health. The system also incorporates advanced programming concepts, such as modular methods for BMI computation and category determination, ensuring both clarity and reliability in operation.

By leveraging object-oriented programming principles, the Enhanced BMI Calculator demonstrates the use of dynamic input handling, conditional logic, and robust functionality. The system emphasizes inclusivity by considering age- and gender-specific health advice, making it suitable for a broad audience. With its intuitive interface and personalized approach, this project aims to promote health awareness and provide users with a practical, efficient tool to support their fitness and wellness journey.

## Objective

**Objective**

The primary objective of this project is to develop a robust and user-friendly BMI Calculator application that caters to diverse user needs by supporting both metric and imperial measurement systems. The system aims to calculate Body Mass Index (BMI) efficiently and classify individuals into standard health categories, such as Underweight, Normal weight, Overweight, or Obesity.

The project seeks to enhance user experience by providing personalized health recommendations based on BMI classification, age, and gender, ensuring relevance and inclusivity. By leveraging object-oriented programming principles, the system demonstrates versatility through modular design, enabling functionality like dynamic input handling, method overriding, and category determination.

This application aspires to empower users with a practical, interactive tool that not only simplifies BMI calculation but also promotes health awareness by offering actionable insights for improving or maintaining their well-being. Through its intuitive interface and comprehensive feature set, the project contributes to fostering a healthier lifestyle for individuals across various demographics.

# 

# CHAPTER 2

# Problem Definition

# The rise of health concerns necessitates tools that provide individuals with quick and accurate insights into their physical well-being. One such critical metric is the Body Mass Index (BMI), which serves as an indicator of whether an individual's weight falls within a healthy range. However, existing BMI calculators often lack personalization, flexibility for different measurement systems, and actionable guidance tailored to individual demographics.

# This project aims to address the need for a comprehensive BMI calculator that not only computes BMI accurately using both metric and imperial units but also categorizes the result into standard health classifications such as Underweight, Normal weight, Overweight, or Obesity. Furthermore, it should offer personalized health recommendations based on the user's age, gender, and BMI category to promote a healthier lifestyle.

# By designing a user-centric, interactive tool, this project seeks to simplify BMI calculation, provide personalized health insights, and encourage individuals to take proactive steps toward maintaining or improving their overall health.

# CHAPTER 3

# METHODOLOGY

# Requirements Analysis:

# Begin by analyzing the need for a BMI calculator that accommodates both metric and imperial measurement systems. Identify user requirements such as calculating BMI, categorizing BMI into health classifications, and providing personalized health recommendations based on age, gender, and BMI category. Understand the need for a user-friendly interface for effective interaction

# System Design:

# Develop a structured design for the BMI calculator. Define the input parameters (age, gender, weight, height) and the necessary outputs (BMI value, category, and recommendations). Plan the logic for handling both metric and imperial units and mapping BMI values to health categories. Outline the flow of user interaction to ensure a smooth process..

# Object-Oriented Programming Implementation:

# Leverage object-oriented programming principles for modular and maintainable code. Implement functions for BMI calculation, category determination, and providing health recommendations. Incorporate concepts such as method overloading (to handle different measurement systems) and encapsulation to maintain clean and organized code.

# User Interface Development:

# Implement an interactive text-based user interface in the main method. Use clear prompts to guide users through the BMI calculation process, including selecting a measurement system, entering their details, and displaying results. Ensure the interface is intuitive and user-friendly.

# Testing and Quality Assurance:

# Conduct comprehensive testing to ensure the system functions as intended. Test cases should include various scenarios, such as different BMI values, measurement system selections, and invalid inputs. Address any bugs or issues encountered during testing to improve system reliability.

# User Interaction and Feedback:

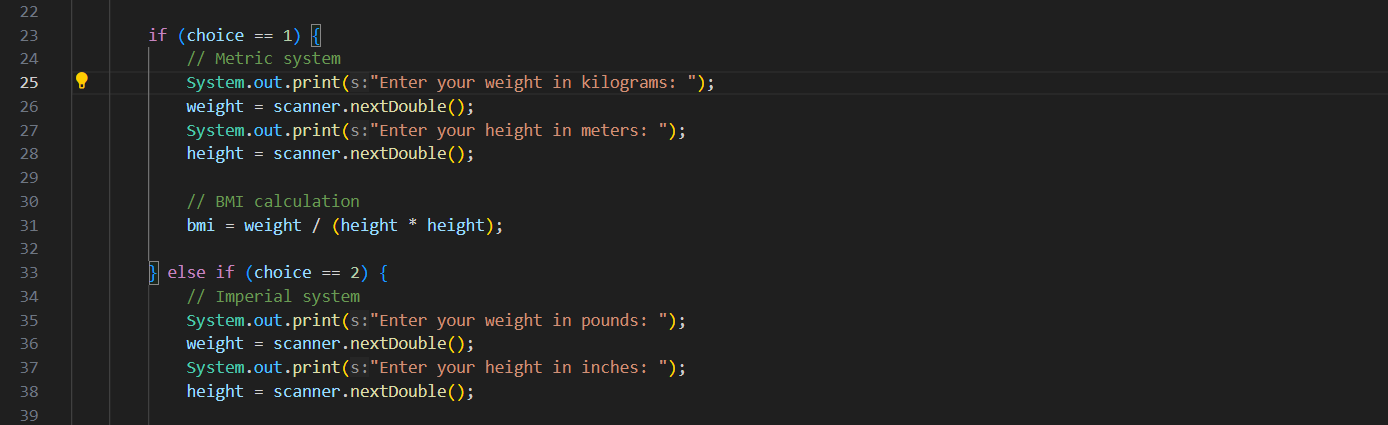
Encourage users to interact with the system and provide feedback on its functionality and user-friendliness. Use feedback to make iterative improvements, ensuring the BMI calculator meets user expectations and effectively fulfills its purpose.

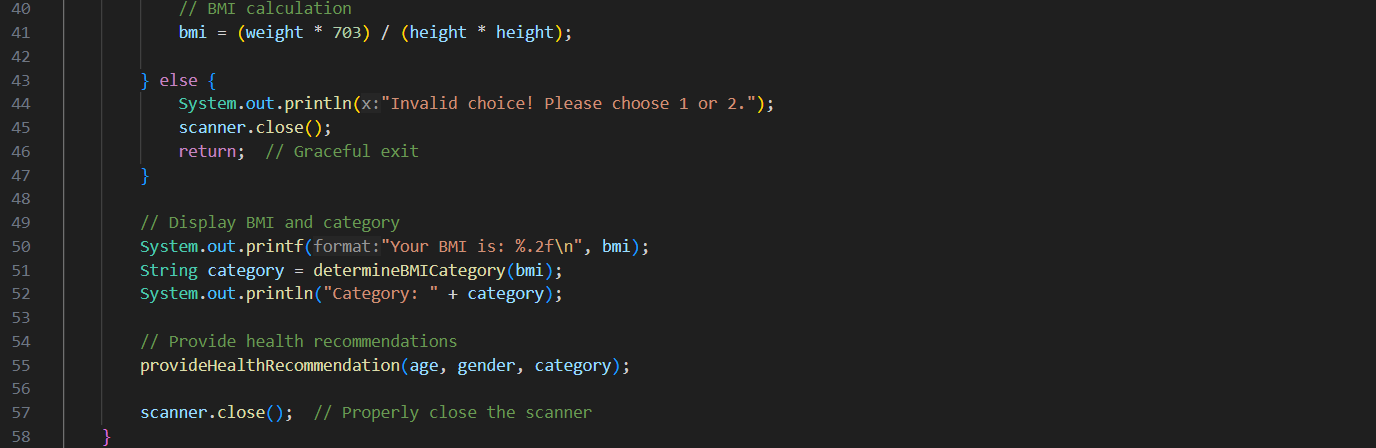
# CHAPTER 4

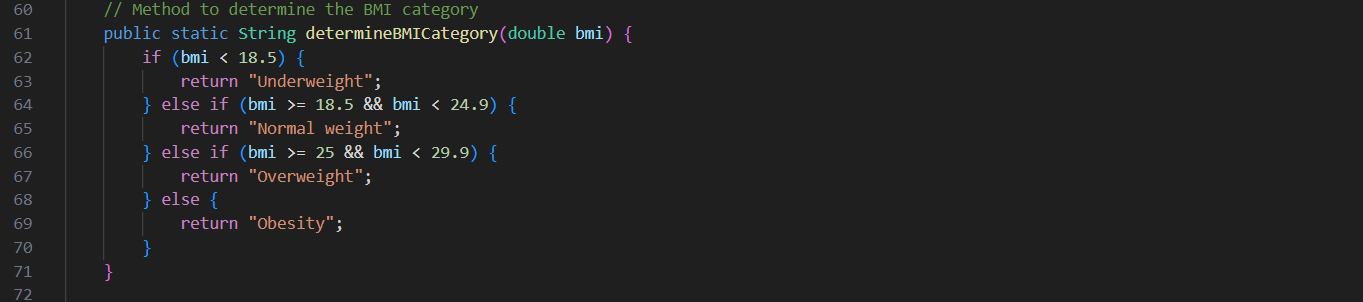
# IMPLEMENTATION OF CODE

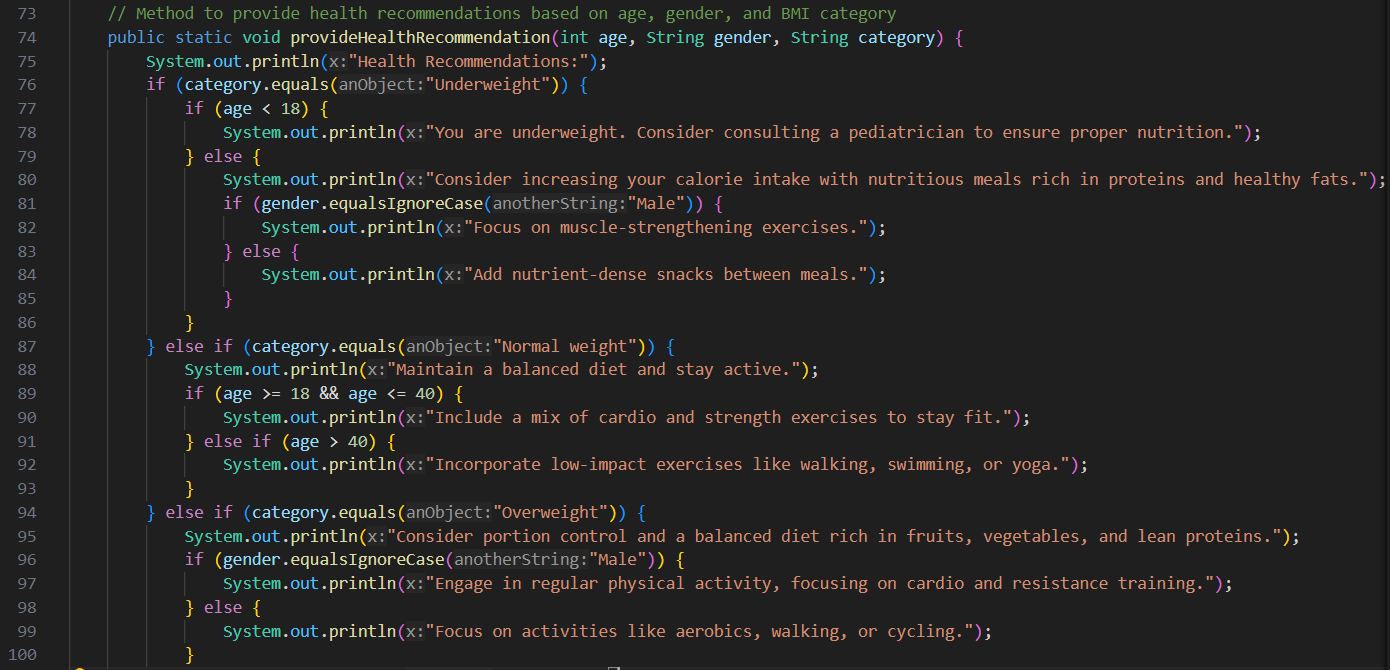
# 

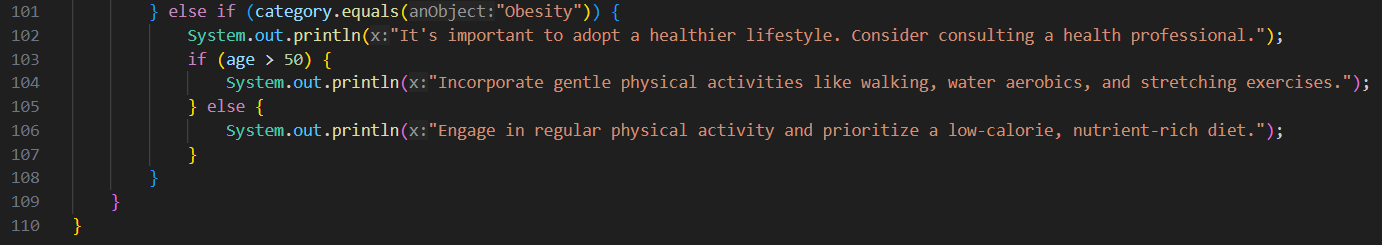
# 





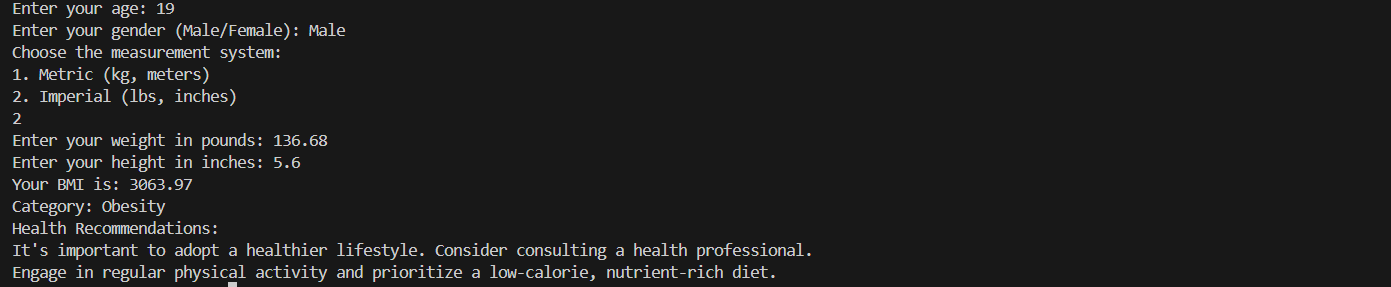
****

****

****

**OUTPUT:**





# 

# FULL CODE

# import java.util.Scanner;

# public class EnhancedBMICalculator {

# public static void main(String[] args) {

# Scanner scanner = new Scanner(System.in);

# // Collect user details

# System.out.print("Enter your age: ");

# int age = scanner.nextInt();

# scanner.nextLine(); // Consume leftover newline

# System.out.print("Enter your gender (Male/Female): ");

# String gender = scanner.nextLine();

# System.out.println("Choose the measurement system:");

# System.out.println("1. Metric (kg, meters)");

# System.out.println("2. Imperial (lbs, inches)");

# int choice = scanner.nextInt();

# double weight, height, bmi = 0;

# if (choice == 1) {

# // Metric system

# System.out.print("Enter your weight in kilograms: ");

# weight = scanner.nextDouble();

# System.out.print("Enter your height in meters: ");

# height = scanner.nextDouble();

# // BMI calculation

# bmi = weight / (height \* height);

# } else if (choice == 2) {

# // Imperial system

# System.out.print("Enter your weight in pounds: ");

# weight = scanner.nextDouble();

# System.out.print("Enter your height in inches: ");

# height = scanner.nextDouble();

# // BMI calculation

# bmi = (weight \* 703) / (height \* height);

# } else {

# System.out.println("Invalid choice! Please choose 1 or 2.");

# scanner.close();

# return; // Graceful exit

# }

# // Display BMI and category

# System.out.printf("Your BMI is: %.2f\n", bmi);

# String category = determineBMICategory(bmi);

# System.out.println("Category: " + category);

# // Provide health recommendations

# provideHealthRecommendation(age, gender, category);

# scanner.close(); // Properly close the scanner

# }

# // Method to determine the BMI category

# public static String determineBMICategory(double bmi) {

# if (bmi < 18.5) {

# return "Underweight";

# } else if (bmi >= 18.5 && bmi < 24.9) {

# return "Normal weight";

# } else if (bmi >= 25 && bmi < 29.9) {

# return "Overweight";

# } else {

# return "Obesity";

# }

# }

# // Method to provide health recommendations based on age, gender, and BMI category

# public static void provideHealthRecommendation(int age, String gender, String category) {

# System.out.println("Health Recommendations:");

# if (category.equals("Underweight")) {

# if (age < 18)

# {

# System.out.println("You are underweight. Consider consulting a pediatrician to ensure proper nutrition.");

# } else {

# System.out.println("Consider increasing your calorie intake with nutritious meals rich in proteins and healthy fats.");

# if (gender.equalsIgnoreCase("Male")) {

# System.out.println("Focus on muscle-strengthening exercises.");

# } else {

# System.out.println("Add nutrient-dense snacks between meals.");

# }

# }

# } else if (category.equals("Normal weight")) {

# System.out.println("Maintain a balanced diet and stay active.");

# if (age >= 18 && age <= 40) {

# System.out.println("Include a mix of cardio and strength exercises to stay fit.");

# } else if (age > 40) {

# System.out.println("Incorporate low-impact exercises like walking, swimming, or yoga.");

# }

# } else if (category.equals("Overweight")) {

# System.out.println("Consider portion control and a balanced diet rich in fruits, vegetables, and lean proteins.");

# if (gender.equalsIgnoreCase("Male")) {

# System.out.println("Engage in regular physical activity, focusing on cardio and resistance training.");

# } else {

# System.out.println("Focus on activities like aerobics, walking, or cycling.");

# }

# } else if (category.equals("Obesity")) {

# System.out.println("It's important to adopt a healthier lifestyle. Consider consulting a health professional.");

# if (age > 50) {

# System.out.println("Incorporate gentle physical activities like walking, water aerobics, and stretching exercises.");

# } else {

# System.out.println("Engage in regular physical activity and prioritize a low-calorie, nutrient-rich diet.");

# }

# }

# }

# }

# OUTPUT:

# Enter your age: 20

# Enter your gender (Male/Female): Female

# Choose the measurement system:

# 1. Metric (kg, meters)

# 2. Imperial (lbs, inches)

# 1

# Enter your weight in kilograms: 52

# Enter your height in meters: 1.61

# Your BMI is: 20.06

# Category: Normal weight

# Health Recommendations:

# Maintain a balanced diet and stay active.

# Include a mix of cardio and strength exercises to stay fit.

# Enter your age: 19

# Enter your gender (Male/Female): Male

# Choose the measurement system:

# 1. Metric (kg, meters)

# 2. Imperial (lbs, inches)

# 2

# Enter your weight in pounds: 136.68

# Enter your height in inches: 5.6

# Your BMI is: 3063.97

# Category: Obesity

# Health Recommendations:

# It's important to adopt a healthier lifestyle. Consider consulting a health professional.

# Engage in regular physical activity and prioritize a low-calorie, nutrient-rich diet.

# Enter your age: 12

# Enter your gender (Male/Female): Female

# Choose the measurement system:

# 1. Metric (kg, meters)

# 2. Imperial (lbs, inches)

# 1

# Enter your weight in kilograms: 30

# Enter your height in meters: 1.52

# Your BMI is: 12.98

# Category: Underweight

# Health Recommendations:

# You are underweight. Consider consulting a pediatrician to ensure proper nutrition.

# CHAPTER 5

# TECHNOLOGY USED

# The provided code is written in Java, a versatile, object-oriented programming language that is widely used for various applications, including desktop, web, and mobile development. Here's a breakdown of the technology aspects used in the code:

# Java Programming Language:

# The entire codebase is written in Java, known for its portability, readability, and ease of development. Java is often chosen for its platform independence, making it suitable for developing cross-platform applications.

# Object-Oriented Programming (OOP):

# The code employs OOP concepts, such as encapsulation, modular design, and the use of separate methods for specific functionalities (e.g., BMI calculation, category determination, health recommendations).

# This structure enhances code readability, reusability, and maintainability.

# Scanner Class:

# The Scanner class from the java.util package is used for capturing user input from the console.

# This facilitates interactive communication with the user, enabling real-time data entry for age, gender, weight, and height.

# Console Output:

# The System.out.printf and System.out.println methods are used for formatted and standard text output to the console.

# These methods provide a simple and effective way to display results and prompt messages to users.

# Conditional Logic (if-else Statements):

# The program uses conditional statements to categorize BMI values and generate health recommendations based on multiple factors like BMI, age, and gender.

# This dynamic control flow ensures tailored results and enhances user experience.

# Mathematical Operations:

# The code performs arithmetic operations to compute BMI using different formulas for metric and imperial units.

# This highlights Java's capability to handle numerical computations efficiently.

# Method Modularity:

# Modular methods like determineBMICategory() and provideHealthRecommendation() break the functionality into logical units.

# This structure follows the best practices for clean and maintainable code.

**Input Validation**:

* Basic input validation ensures that users select valid options (e.g., choosing between metric and imperial systems) and gracefully handles invalid choices.
* Although minimal, this serves as a foundation for adding more robust validation features.



# CHAPTER 6

# RESULT AND ANALYSIS

**Code Execution Result:**

* The program prompts the user to input their age, gender, and choice of measurement system (metric or imperial).
* Depending on the choice, the user enters their weight and height in the corresponding units.
* The program calculates the Body Mass Index (BMI) using the appropriate formula for the chosen measurement system.
* The BMI value is categorized into one of four standard health classifications: Underweight, Normal weight, Overweight, or Obesity.
* Personalized health recommendations are displayed based on the user's age, gender, and BMI category.
* The program outputs all information to the console in a structured and user-friendly format.

**Analysis:**

* **Object-Oriented Design:**
* The code employs a modular structure with separate methods for BMI calculation, health category determination, and personalized recommendations.
* The use of encapsulation ensures that individual functionalities are logically separated, making the program easier to maintain and extend.

**Flexibility and Customization:**

* The system supports both metric and imperial measurement systems, enhancing its adaptability for a global audience.
* Conditional logic enables tailored health recommendations based on multiple factors such as BMI, age, and gender.

**BMI Calculation and Health Categorization:**

* The code uses standard formulas for BMI calculation and categorizes results based on established health ranges, ensuring accuracy and reliability.
* The output provides a clear indication of the user's health status with actionable insights for improvement.

**User Interaction:**

* Interactive input is managed using the Scanner class, allowing users to provide their details easily.

# The interface includes clear prompts and structured outputs, ensuring a smooth user experience. However, the text-based interface could be improved with a graphical or web-based interface for enhanced usability.

**Educational Context:**

* The code serves as an excellent learning example, demonstrating fundamental programming concepts such as modular methods, conditional statements, and input handling in Java.
* It illustrates the practical application of health-related calculations and offers insights into real-world programming scenarios**.**

**Error Handling:**

# Basic error handling is implemented to manage invalid menu choices and ensure program stability.

# Additional validation for inputs like negative or non-numeric values could improve the system's robustness.

# CHAPTER 7

# CONCLUSION

# The implemented BMI calculator in Java demonstrates a practical application of object-oriented principles to solve a real-world problem. The code is well-structured, leveraging modularity through separate methods for BMI calculation, category determination, and personalized health recommendations. The flexibility of handling both metric and imperial measurement systems ensures adaptability for users with different preferences. The use of clear prompts and interactive input via the Scanner class enhances user experience, while the inclusion of tailored health advice based on age, gender, and BMI category adds a personalized touch to the system.

# Although the current implementation is robust and educational, there is potential for further improvements, such as adding a graphical user interface, enhancing input validation, and incorporating more comprehensive health insights. The simplicity of the code makes it an excellent example for learning Java and understanding fundamental programming concepts like encapsulation and conditional logic. Overall, the BMI calculator serves as a functional, user-friendly tool with ample scope for future enhancements and feature integrations.