NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

A MINI PROJECT REPORT

for

Mini Project in Java (19CSE48)

on

CALORIE TRACKER

Submitted by

B.AJAY
USN: 1NH20CS039, Sem-Sec: 4-A

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Academic Year: 2021-22(EVEN SEM)

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

CERTIFICATE

This is to certify that the mini project work titled

CALORIE TRACKER

submitted in partial fulfillment of the degree of Bachelor of Engineering in Computer Science and Engineering by

B.AJAY
USN:1NH20CS039

DURING

EVEN SEMESTER 2021-2022

for

Course: Mini Project in Java-19CSE48

Signature of Reviewer Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner	*Signature with date*
1.____________________	____________________
2.____________________	____________________

ABSTRACT

Having an efficient and useful fitness care system is an important aspect by which a country’s health can be classified upon. Nowadays , fitness has become a sector which has seen large growth in terms of both employment and revenue.

The Corona virus pandemic has also taught the world about the importance of fitness that have played a key role in this battle against the virus. This mini-project helps users to maintain their fitness level and overall health through the development of a desktop application.

Calorie tracker is a mini project which is an all round health application that is user interactive and It comes with many features such as workout companion,weight gain/loss companion,calories burnt calculator and BMI calculator.

This mini project makes the process of calculating calories as easy as possible. Keeping manual count of the calories consumed or burned can be a very time consuming and tedious process. This project is designed to provide the user with timed workout sessions of various difficulty levels. It provides diet and workout plans according to their Body Mass Index (BMI). This project can also be used to keep track of the amount of calories burnt after a physical activity.

The entire program has been developed in Java and uses the Eclipse IDE for running the Java application. The mini-project is completely based on the high-level language, Java to provide a simple and easy to understand platform for the users.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned my efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman, New Horizon Educational Institutions, for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha,** Principal, New Horizon College of Engineering, for his constant support and encouragement.

I would like to thank **Dr. Anandhi R J**, Professor and Dean-Academics, NHCE, for her valuable guidance.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and HOD, Department of Computer Science and Engineering, for her constant support.

I also express my gratitude to Ms. **Jeevitha** , Professor, Department of Computer Science and Engineering,my mini project reviewer, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

B.Ajay

USN: 1NH20CS039

CONTENTS

ABSTRACT **I**

ACKNOWLEDGEMENT **II**

LIST OF FIGURES **V**

1. INTRODUCTION

1.1. PROBLEM DEFINITION **1**

1.2. OBJECTIVES **1**

1.3. METHODOLOGY TO BE FOLLOWED **2**

1.4. EXPECTED OUTCOMES **2**

1.5. HARDWARE AND SOFTWARE REQUIREMENTS **3**

2. **FUNDAMENTALS OF JAVA PROGRAMMING**

2.1. INTRODUCTION **4**

2.2. CLASS **4**

2.3. OBJECT **5**

2.4. CONSTRUCTORS **5**

2.5. VARIABLES **6**

2.6. JAVA PACKAGES **7**

2.7. INHERITANCE **8**

2.8. POLYMORPHISM **10**

2.9. ENCAPSULATION **11**

2.10. EXCEPTION HANDLING **12**

3. DESIGN

3.1. DESIGN GOALS **15**

3.2. PSEUDOCODE **15**

3.3. FLOWCHART **16**

4. IMPLEMENTATION

4.1. MODULE 1 FUNCTIONALITY **17**

4.2. MODULE 2 FUNCTIONALITY **17**

4.3. MODULE 3 FUNCTIONALITY **19**

4.4. MODULE 4 FUNCTIONALITY **19**

4.5. MODULE 5 FUNCTIONALITY **20**

4.6. MODULE 6 FUNCTIONALITY **21**

4.7. MODULE 7 FUNCTIONALITY **22**

5. RESULTS

5.1. SNAPSHOTS **24**

6. CONCLUSION **31**

REFERENCES **32**

LIST OF FIGURES

Figure No	Figure Description	Page No
2.1	Example dog object	5
2.2	Variables	6
2.3	Packages	7
2.4	Single Inheritance	8
2.5	Multilevel Inheritance	9
2.6	Hierarchical Inheritance	9
2.7	Polymorphism	11
2.8	Encapsulation	11
2.9	Types of Exceptions	13
3.1	Flowchart	16
4.1	Activity Class	17
4.2	Workout Companion method	17
4.3	Workout Companion method	18
4.4	Workout Companion method	18
4.5	Calorie Suggestion method	19
4.6	Calorie Suggestion method	19
4.7	Calories burnt method	20
4.8	Calories burnt method	20
4.9	Calories burnt method	21

4.10	BMI method	21
4.11	BMI method	22
4.12	Main method	22
4.13	Main method	23
5.1	Main menu	24
5.2	Workout companion	24
5.3	Workout companion	25
5.4	Calories consumed	25
5.5	Calories consumed	26
5.6	Calorie suggestion	26
5.7	Calorie suggestion	27
5.8	BMI calculator	27
5.9	BMI calculator	28
5.10	Calories burnt	28
5.11	Calories burnt	29

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

Manually noting down the food items consumed and exercises performed in order to keep track of the calories can be a very time consuming and tedious process.This project is designed to computerize this whole process so that the user doesn’t need to do it manually.Manually calculating the calories required for weight loss or gain is very complicated.With the help of this mini project this problem can also be solved as it can instantky calculate an provide the values to the user. It also gives tips for the user to follow during their weight loss/gain journey.It also comes with a workout companion which runs a timer for each exercise so that the user can follow along and perform the exercises.This project makes the process of Calculating BMI very easy and it also provides customised diet and workout plans to the user according to their BMI values.

1.2 OBJECTIVES

- To make calorie tracking an easy task.
- To give the user tips to follow during weight loss/gain.
- To organise customised workout sessions.
- To Provide customised workout and diet plans.
- To help in calculating BMI(Body Mass Index).

1.3 METHODOLOGY TO BE FOLLOWED

Workout companion method - for the users to follow the exercises through a timer which runs for each exercise.

Calorie consumed counter method – for receiving the input of all food items consumed and calculating the total calories consumed and to determine if the calorie goal is reached or not.

Weight gain/loss method – For calculating and display the the amount of calories the user needs to consume to put weight or cut down to reduce weight.

BMI calculator method – To calculate the BMI ad display diet plans and workout plans according to the result.

Calorie burnt counter method – To receive all the physical activities performed by the user and calculate and display the total amount of calories burnt.

1.4 EXPECTED OUTCOMES

- To calculate and display the total amount of calories consumed.
- To display a timer for each exercise and simulate a workout session .
- To calculate and display the amount of calories needed for weight gain or loss.
- To calculate BMI and provide diet plans and workout plans accordingly.
- To calculate and display the total amount of calories burnt.

1.5 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Requirements :

1) RAM – 3GB OR MORE

2) Processor – INTEL CORE i3

Software Requirements :

1) Eclipse IDE

2) Operating system – Windows 11

CHAPTER 2

FUNDAMENTALS OF JAVA PROGRAMMING

2.1 Introduction

JAVA was developed by James Gosling at Sun Microsystems Inc in the year 1995, later acquired by Oracle Corporation. It is a simple programming language. Java makes writing, compiling, and debugging programming easy. It helps to create reusable code and modular programs. Java is a class-based, object-oriented programming language and is designed to have as few implementation dependencies as possible. A general-purpose programming language made for developers to write once run anywhere that is compiled Java code can run on all platforms that support Java. Java applications are compiled to byte code that can run on any Java Virtual Machine. The syntax of Java is similar to c/c++.

2.2 Class

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

A class in Java can contain:

- Fields
- Methods
- Constructors
- Blocks
- Nested class and interface

2.3 Object

An entity that has state and behavior is known as an object e.g., chair, bike, marker, pen, table, car, etc. It can be physical or logical (tangible and intangible). The example of an intangible object is the banking system.

An object has three characteristics:

- State **:** represents the data (value) of an object.
- Behavior **:** represents the behavior (functionality) of an object such as deposit, withdraw, etc.
- Identity **:** An object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. However, it is used internally by the JVM to identify each object uniquely.

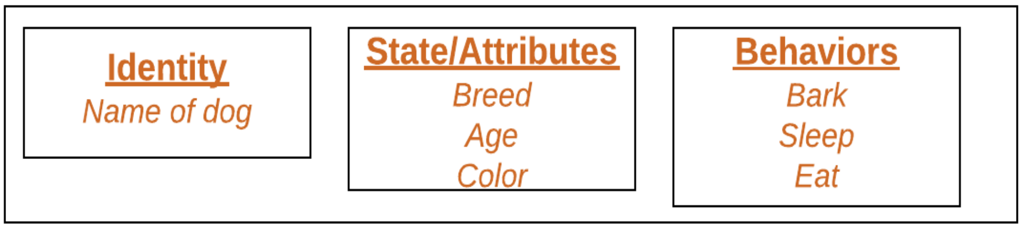


Fig 2.1 Example dog Object

2.4 Constructors

When discussing about classes, one of the most important sub topic would be constructors. Every class has a constructor. If we do not explicitly write a constructor for a class, the Java compiler builds a default constructor for that class.

Each time a new object is created, at least one constructor will be invoked. The main rule of constructors is that they should have the same name as the class. A class can have more than one constructor.

2.5 Variables

Variable in Java is a data container that saves the data values during Java program execution. Every variable is assigned a data type that designates the type and quantity of value it can hold. A variable is a memory location name for the data.

A variable is a name given to a memory location. It is the basic unit of storage in a program.

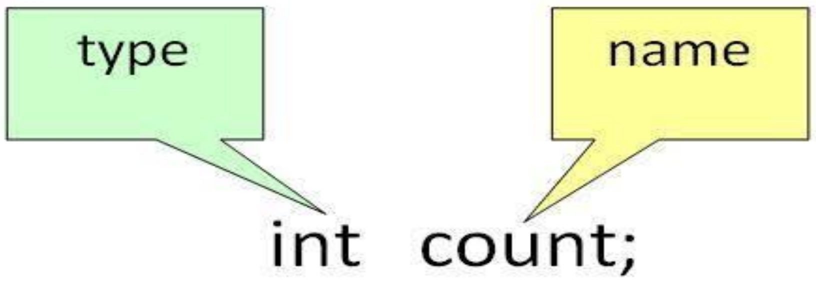


Fig 2.2 Variables

Now let us discuss different types of variables which are listed as follows :

1. Local Variables
2. Instance Variables
3. Static Variables

Local Variables : A variable defined within a block or method or constructor is called a local variable.

Instance Variables : Instance variables are non-static variables and are declared in a class outside of any method, constructor, or block.

Static Variables : Static variables are also known as class variables.

2.6 JAVA Packages

Package in Java is a mechanism to encapsulate a group of classes, sub packages and interfaces.

Packages are used for :

- Preventing naming conflicts
- Providing controlled access

Packages can be considered as data encapsulation (or data-hiding).

A package is a container of a group of related classes where some of the classes are accessible are exposed and others are kept for internal purpose.

We can reuse existing classes from the packages as many time as we need it in our program.

There are 2 types of packages in JAVA :

1)In-built packages

2)User-defined packages

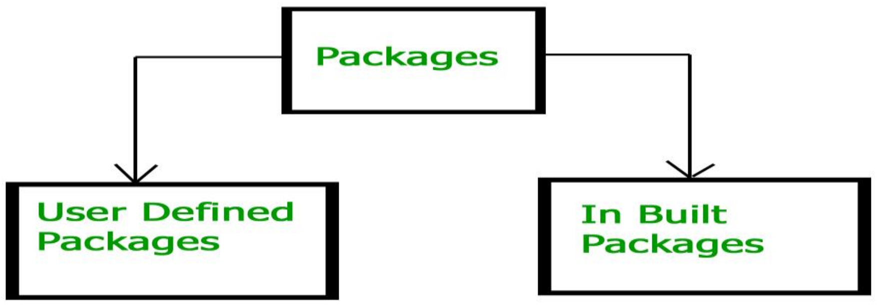


Fig 2.3 Packages

2.7 Inheritance

Inheritance is an important pillar of OOP(Object-Oriented Programming). It is the mechanism in java by which one class is allowed to inherit the features(fields and methods) of another class.

Important terminology :

Super Class: The class whose features are inherited is known as superclass(or a base class or a parent class).

Sub Class: The class that inherits the other class is known as a subclass(or a derived class, extended class, or child class). The subclass can add its own fields and methods in addition to the superclass fields and methods.

Reusability: Inheritance supports the concept of "reusability", i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class.

Types of Inheritance in Java

Below are the different types of inheritance which are supported by Java.

1. Single Inheritance: In single inheritance, subclasses inherit the features of one superclass. In the image below, class A serves as a base class for the derived class B.

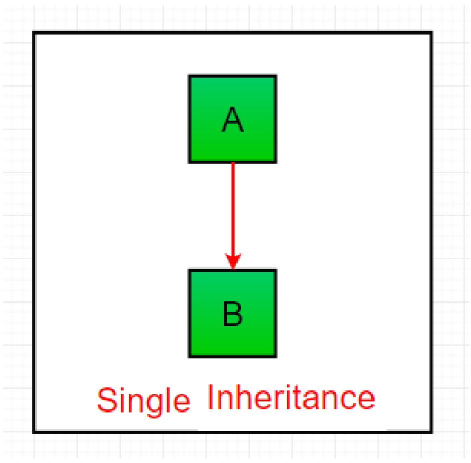


Fig 2.4 Single inheritance

2. Multilevel Inheritance: In Multilevel Inheritance, a derived class will be inheriting a base class and as well as the derived class also act as the base class to other class. In the below image, class A serves as a base class for the derived class B, which in turn serves as a base class for the derived class C. In Java, a class cannot directly access the grandparent’s members.

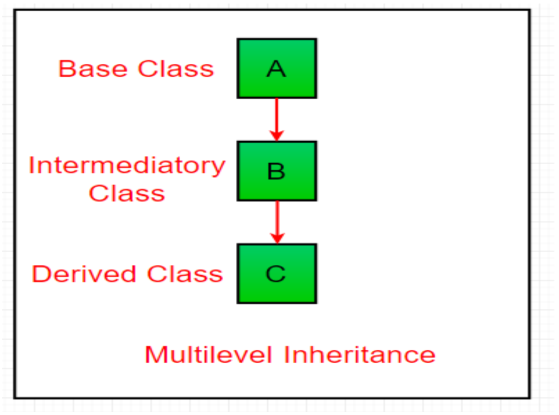


Fig 2.5 Multilevel Inheritance

3. Hierarchical Inheritance: In Hierarchical Inheritance, one class serves as a superclass (base class) for more than one subclass. In the below image, class A serves as a base class for the derived class B, C and D.

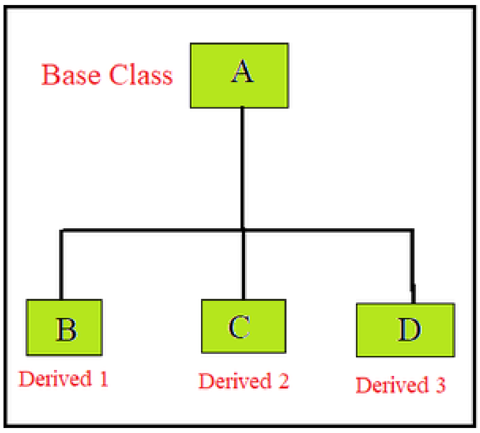


Fig 2.6 Hierarchical Inheritance

2.8 Polymorphism

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form.

Polymorphism is considered one of the important features of Object-Oriented Programming.

In Java polymorphism is mainly divided into two types:

1) Compile-time Polymorphism

2) Runtime Polymorphism

Type 1: Compile-time polymorphism

It is also known as static polymorphism. This type of polymorphism is achieved by function overloading or operator overloading.

Method Overloading : When there are multiple functions with the same name but different parameters then these functions are said to be overloaded. Functions can be overloaded by change in the number of arguments or/and a change in the type of arguments.

Type 2: Runtime polymorphism

It is also known as Dynamic Method Dispatch. It is a process in which a function call to the overridden method is resolved at Runtime. This type of polymorphism is achieved by Method Overriding. Method overriding, on the other hand, occurs when a derived class has a definition for one of the member functions of the base class. That base function is said to be overridden.

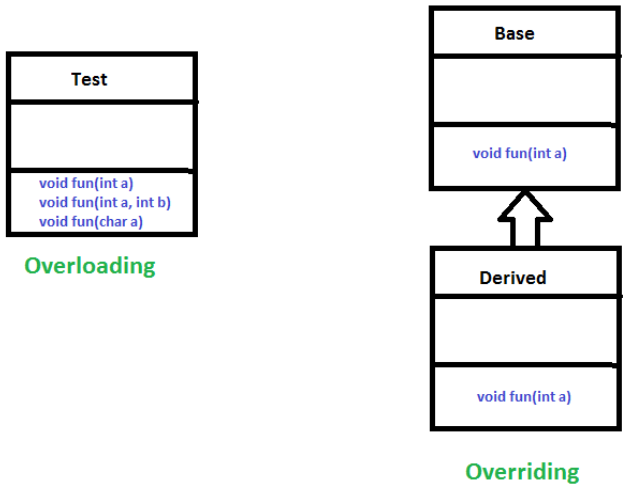


Fig 2.7 Polymorphism

2.9 Encapsulation

Encapsulation is defined as the wrapping up of data under a single unit. Another way to think about encapsulation is, that it is a protective shield that prevents the data from being accessed by the code outside this shield.

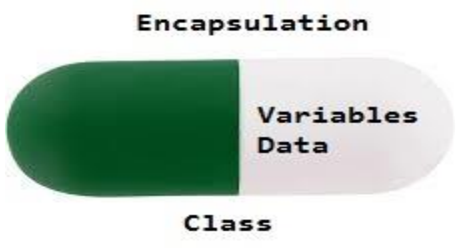


Fig 2.8 Encapsulation

Technically in encapsulation, the variables or data of a class is hidden from any other class and can be accessed only through any member function of its own class in which it is declared.

Encapsulation can be achieved by Declaring all the variables in the class as private and writing public methods in the class to set and get the values of variables.

Advantages of Encapsulation:

Data Hiding: it is a way of restricting the access of our data members by hiding the implementation details. Encapsulation also provides a way for data hiding. The user will have no idea about the inner implementation of the class. It will not be visible to the user how the class is storing values in the variables. The user will only know that we are passing the values to a setter method and variables are getting initialized with that value.

Increased Flexibility: We can make the variables of the class read-only or write-only depending on our requirement. If we wish to make the variables read-only then we have to omit the setter methods.

Reusability: Encapsulation also improves the re-usability and is easy to change with new requirements.

Testing code is easy: Encapsulated code is easy to test for unit testing.

2.10 Exception Handling

Exception Handling in Java is one of the effective means to handle the runtime errors so that the regular flow of the application can be preserved.

Exception is an unwanted or unexpected event, which occurs during the execution of a program, i.e. at run time, that disrupts the normal flow of the program's instructions. Exceptions can be caught and handled by the program. When an exception occurs within a method, it creates an object. This object is called the exception object. It contains information about the exception, such as the name and description of the exception and the state of the program when the exception occurred.

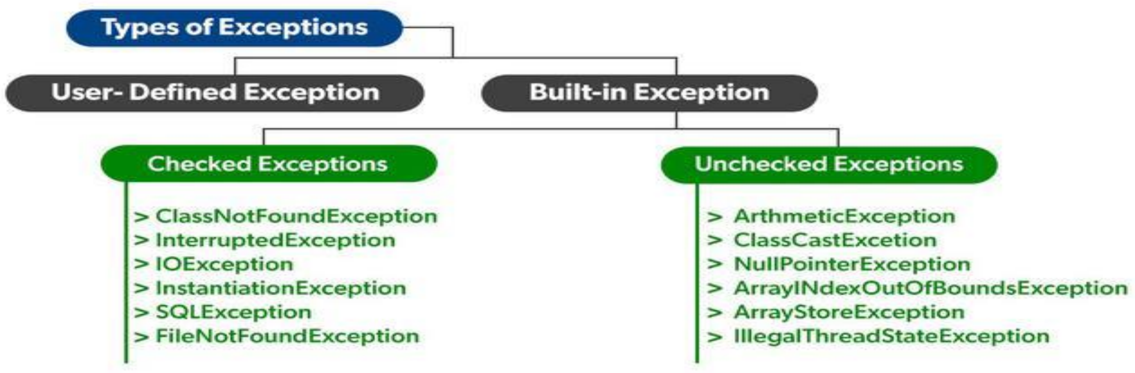


Fig 2.9 Types of exceptions

Types of Exceptions :

Exceptions can be categorized in two ways:

1) Built-in Exceptions
2) Checked Exception
3) Unchecked Exception
4) User-Defined Exceptions

Let us discuss the above-defined listed exception that is as follows:

1. Built-in Exceptions :

Built-in exceptions are the exceptions that are available in Java libraries. These exceptions are suitable to explain certain error situations.

2.Checked Exceptions :

Checked exceptions are called compile-time exceptions because these exceptions are checked at compile-time by the compiler.

3.Unchecked Exceptions:

The unchecked exceptions are just opposite to the checked exceptions. The compiler will not check these exceptions at compile time. In simple words, if a program throws an unchecked exception, and even if we didn’t handle or declare it, the program would not give a compilation error.

4. User-Defined Exceptions:

Sometimes, the built-in exceptions in Java are not able to describe a certain situation. In such cases, users can also create exceptions, which are called ‘user-defined Exceptions’.

The advantages of Exception Handling in Java are as follows :

- Provision to Complete Program Execution
- Easy Identification of Program Code and Error-Handling Code
- Propagation of Errors
- Meaningful Error Reporting
- Identifying Error Types

CHAPTER 3

DESIGN

3.1 DESIGN GOALS

Our design mechanism includes two main things:

1. Reduced complexity in keeping track of calories.
2. A user friendly interface to maintain one's fitness.
3. To provide diet plans according to the user's BMI.
4. Arranging timed workout sessions .

3.2 ALGORITHM /PSEUDOCODE

```
Step 1 : Start
Step 2 : Ask user which feature of the program they want to use .
Step 3 : run the timer for each exercise throughout the workout session and then display the amount of calories burnt during the session.
Step 4 : ask the necessary details from the user and calculate and display if their calorie goal has been reached or not.
Step 5 : Ask the necessary details from the user and then calculate and display the number of calories that the user requires to put or cut weight.
Step 6 : Ask the necessary details from the user and then calculate and display the BMI of the user along with diet and workout plan suggestions
Step 7 : Ask the necessary details from the user and then display the total amount of calories he/she have burnt after the physical activities.
Step 8 : End
```

3.3 FLOWCHART

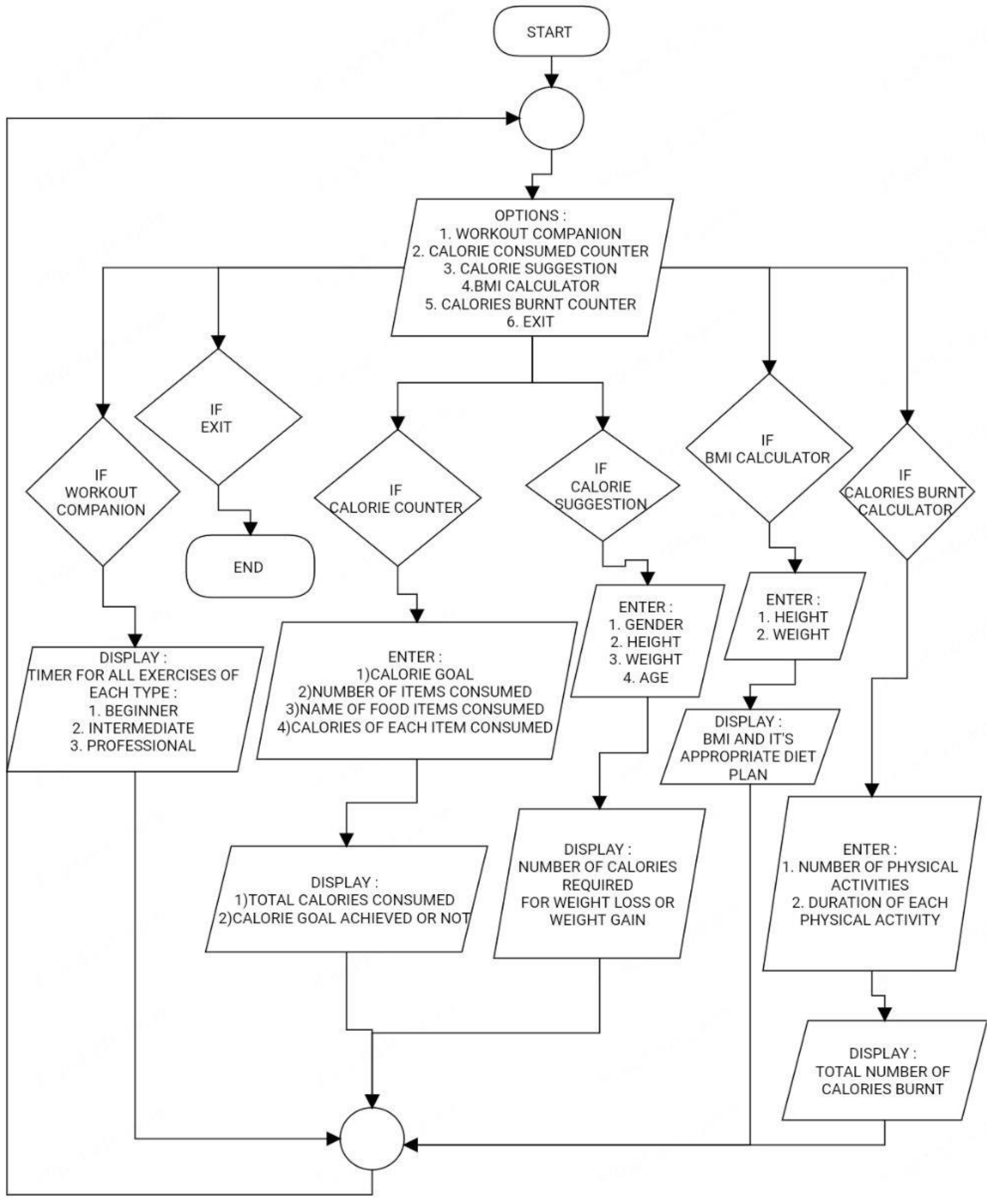
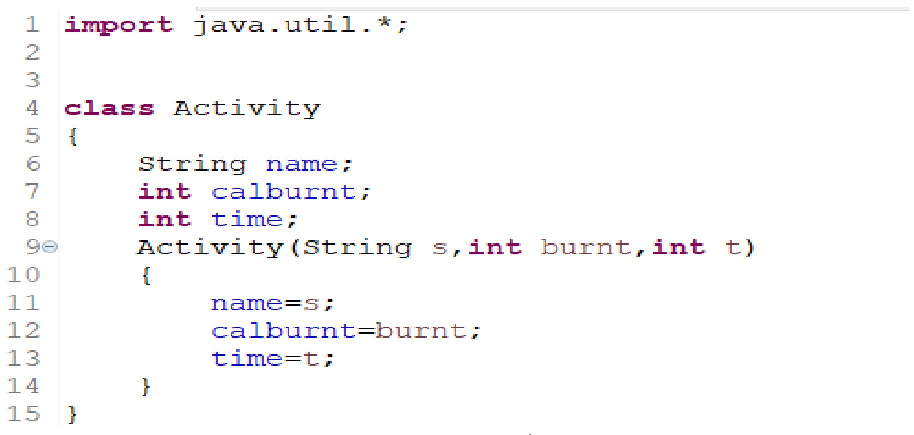

START
OPTIONS :
1. WORKOUT COMPANION
2. CALORIE CONSUMED COUNTER
3. CALORIE SUGGESTION
4.BMI CALCULATOR
5. CALORIES BURNT COUNTER
6. EXIT
IF EXIT
END
IF WORKOUT COMPANION
DISPLAY : TIMER FOR ALL EXERCISES OF EACH TYPE : 1. BEGINNER 2. INTERMEDIATE 3. PROFESSIONAL
IF CALORIE COUNTER
ENTER : 1)CALORIE GOAL 2)NUMBER OF ITEMS CONSUMED 3)NAME OF FOOD ITEMS CONSUMED 4)CALORIES OF EACH ITEM CONSUMED
DISPLAY : 1)TOTAL CALORIES CONSUMED 2)CALORIE GOAL ACHIEVED OR NOT
IF CALORIE SUGGESTION
ENTER : 1. GENDER 2. HEIGHT 3. WEIGHT 4. AGE
DISPLAY : NUMBER OF CALORIES REQUIRED FOR WEIGHT LOSS OR WEIGHT GAIN
IF BMI CALCULATOR
ENTER : 1. HEIGHT 2. WEIGHT
DISPLAY : BMI AND IT'S APPROPRIATE DIET PLAN
IF CALORIES BURNT CALCULATOR
ENTER : 1. NUMBER OF PHYSICAL ACTIVITIES 2. DURATION OF EACH PHYSICAL ACTIVITY
DISPLAY : TOTAL NUMBER OF CALORIES BURNT

Fig 3.1 flowchart

CHAPTER 4

IMPLEMENTATION

4.1 MODULE 1 FUNCTIONALITY

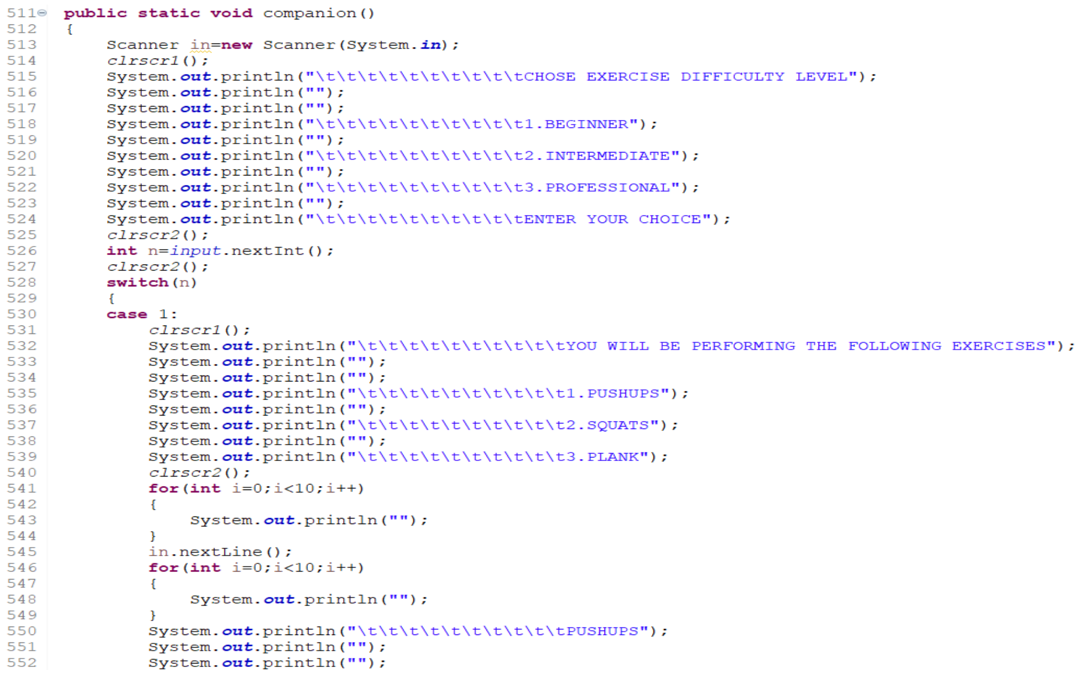


```java
import java.util.*;


class Activity
{
     String name;
     int calburnt;
     int time;
     Activity(String s,int burnt,int t)
     {
          name=s;
          calburnt=burnt;
          time=t;
     }
}
```

Fig 4.1 Activity Class

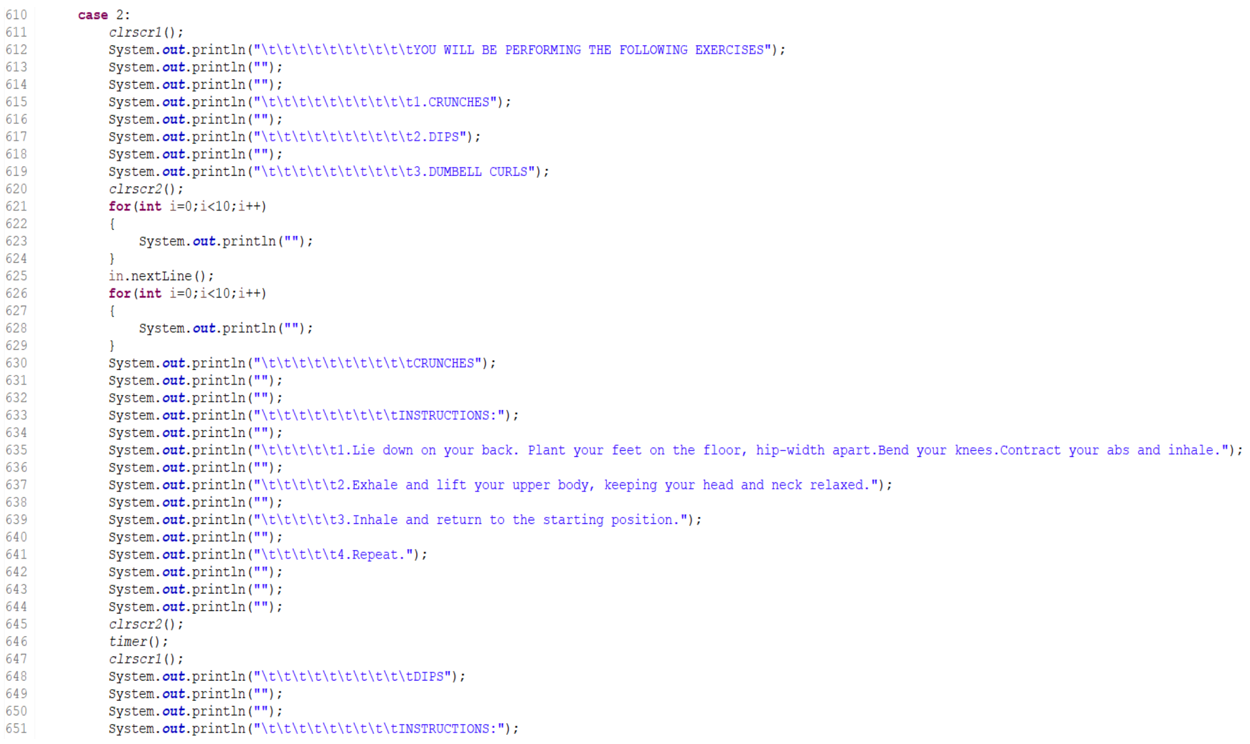
Class Activity is used in the calories burnt method in which an object of this class is created for each physical activity .

4.2 MODULE 2 FUNCTIONALITY



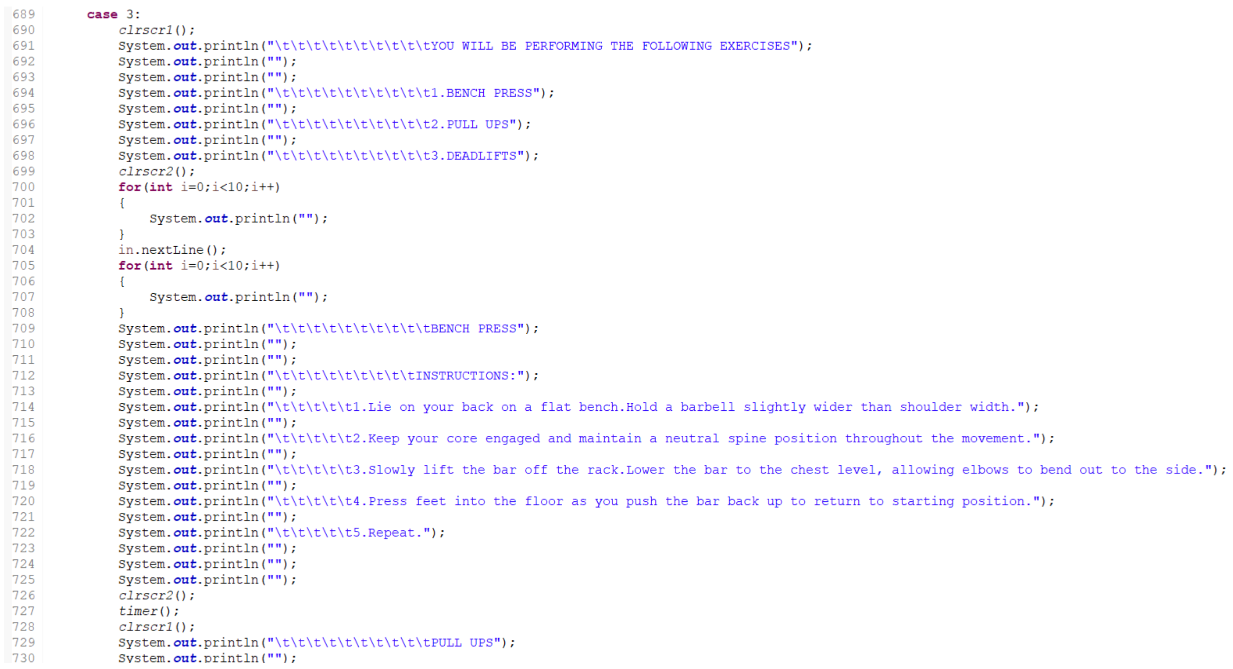
```java
public static void companion()
{
    Scanner in=new Scanner(System.in);
    clrscr1();
    System.out.println("\t\t\t\t\t\t\t\t\t\tCHOSE EXERCISE DIFFICULTY LEVEL");
    System.out.println("");
    System.out.println("");
    System.out.println("\t\t\t\t\t\t\t\t\t\t1.BEGINNER");
    System.out.println("");
    System.out.println("\t\t\t\t\t\t\t\t\t\t2.INTERMEDIATE");
    System.out.println("");
    System.out.println("\t\t\t\t\t\t\t\t\t\t3.PROFESSIONAL");
    System.out.println("");
    System.out.println("\t\t\t\t\t\t\t\t\t\tENTER YOUR CHOICE");
    clrscr2();
    int n=input.nextInt();
    clrscr2();
    switch(n)
    {
    case 1:
        clrscr1();
        System.out.println("\t\t\t\t\t\t\t\t\t\tYOU WILL BE PERFORMING THE FOLLOWING EXERCISES");
        System.out.println("");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\t\t1.PUSHUPS");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\t\t2.SQUATS");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\t\t3.PLANK");
        clrscr2();
        for(int i=0;i<10;i++)
        {
            System.out.println("");
        }
        in.nextLine();
        for(int i=0;i<10;i++)
        {
            System.out.println("");
        }
        System.out.println("\t\t\t\t\t\t\t\t\t\tPUSHUPS");
        System.out.println("");
        System.out.println("");
```

Fig 4.2 Workout companion method



```java
    case 2:
        clrscr1();
        System.out.println("\t\t\t\t\t\t\t\t\t\tYOU WILL BE PERFORMING THE FOLLOWING EXERCISES");
        System.out.println("");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\t\t1.CRUNCHES");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\t\t2.DIPS");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\t\t3.DUMBELL CURLS");
        clrscr2();
        for(int i=0;i<10;i++)
        {
            System.out.println("");
        }
        in.nextLine();
        for(int i=0;i<10;i++)
        {
            System.out.println("");
        }
        System.out.println("\t\t\t\t\t\t\t\t\t\tCRUNCHES");
        System.out.println("");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\tINSTRUCTIONS:");
        System.out.println("");
        System.out.println("\t\t\t\t\t1.Lie down on your back. Plant your feet on the floor, hip-width apart.Bend your knees.Contract your abs and inhale.");
        System.out.println("");
        System.out.println("\t\t\t\t\t2.Exhale and lift your upper body, keeping your head and neck relaxed.");
        System.out.println("");
        System.out.println("\t\t\t\t\t3.Inhale and return to the starting position.");
        System.out.println("");
        System.out.println("\t\t\t\t\t4.Repeat.");
        System.out.println("");
        System.out.println("");
        System.out.println("");
        clrscr2();
        timer();
        clrscr1();
        System.out.println("\t\t\t\t\t\t\t\t\t\tDIPS");
        System.out.println("");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\tINSTRUCTIONS:");
```

Fig 4.3 Workout companion method

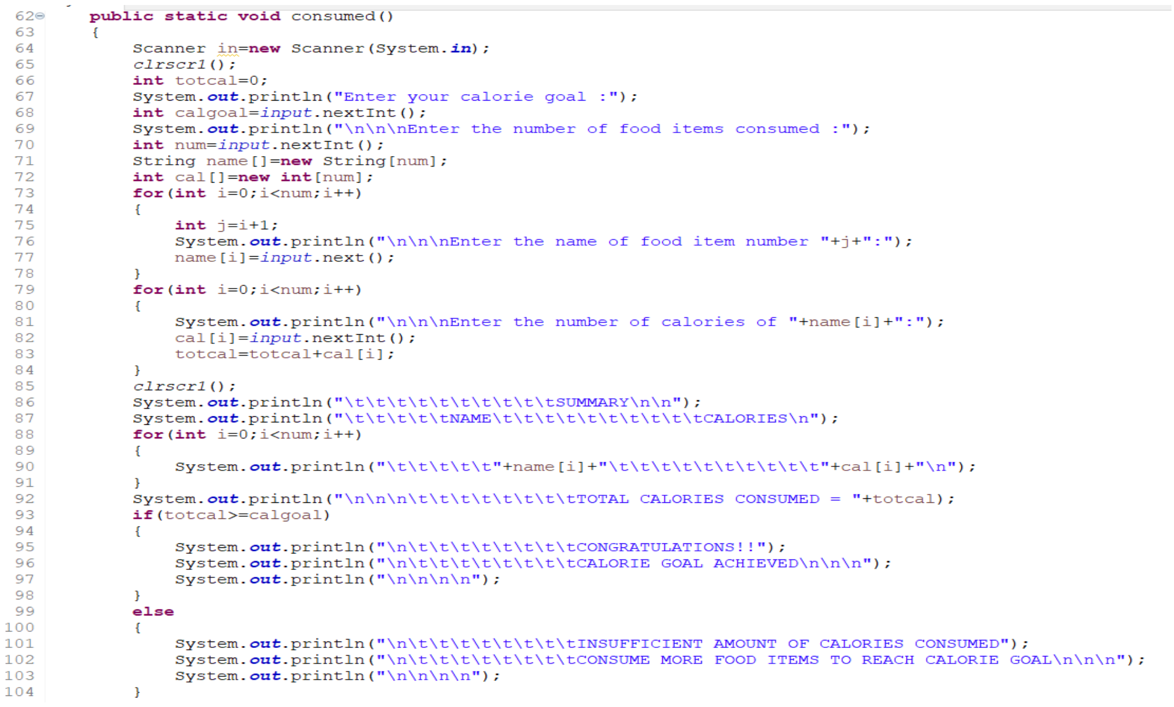


```java
    case 3:
        clrscr1();
        System.out.println("\t\t\t\t\t\t\t\t\t\tYOU WILL BE PERFORMING THE FOLLOWING EXERCISES");
        System.out.println("");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\t\t1.BENCH PRESS");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\t\t2.PULL UPS");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\t\t3.DEADLIFTS");
        clrscr2();
        for(int i=0;i<10;i++)
        {
            System.out.println("");
        }
        in.nextLine();
        for(int i=0;i<10;i++)
        {
            System.out.println("");
        }
        System.out.println("\t\t\t\t\t\t\t\t\t\tBENCH PRESS");
        System.out.println("");
        System.out.println("");
        System.out.println("\t\t\t\t\t\t\t\t\tINSTRUCTIONS:");
        System.out.println("");
        System.out.println("\t\t\t\t\t1.Lie on your back on a flat bench.Hold a barbell slightly wider than shoulder width.");
        System.out.println("");
        System.out.println("\t\t\t\t\t2.Keep your core engaged and maintain a neutral spine position throughout the movement.");
        System.out.println("");
        System.out.println("\t\t\t\t\t3.Slowly lift the bar off the rack.Lower the bar to the chest level, allowing elbows to bend out to the side.");
        System.out.println("");
        System.out.println("\t\t\t\t\t4.Press feet into the floor as you push the bar back up to return to starting position.");
        System.out.println("");
        System.out.println("\t\t\t\t\t5.Repeat.");
        System.out.println("");
        System.out.println("");
        System.out.println("");
        clrscr2();
        timer();
        clrscr1();
        System.out.println("\t\t\t\t\t\t\t\t\t\tPULL UPS");
        System.out.println("");
```

Fig 4.4 Workout companion method

In this method,the program displays various exercises of different difficulty levels such as beginner,intermediate,professional and it displays a timer of 10 seconds for each exercise.

4.3 MODULE 3 FUNCTIONALITY

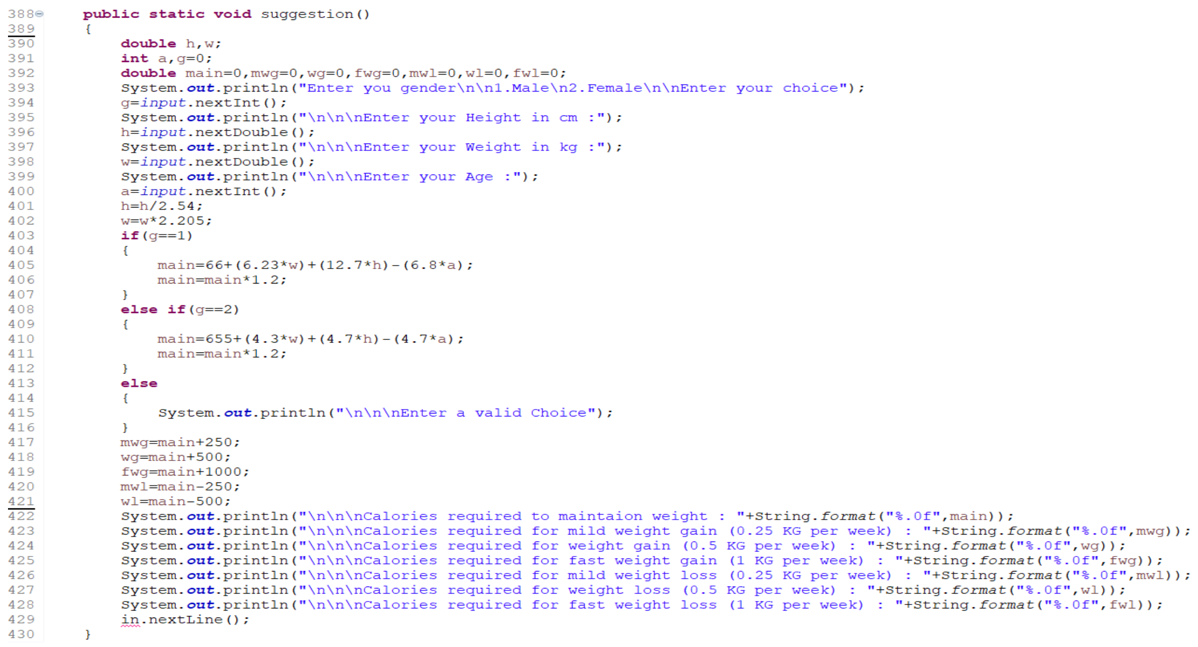


```java
public static void consumed()
{
    Scanner in=new Scanner(System.in);
    clrscr1();
    int totcal=0;
    System.out.println("Enter your calorie goal :");
    int calgoal=input.nextInt();
    System.out.println("\n\n\nEnter the number of food items consumed :");
    int num=input.nextInt();
    String name[]=new String[num];
    int cal[]=new int[num];
    for(int i=0;i<num;i++)
    {
        int j=i+1;
        System.out.println("\n\n\nEnter the name of food item number "+j+":");
        name[i]=input.next();
    }
    for(int i=0;i<num;i++)
    {
        System.out.println("\n\n\nEnter the number of calories of "+name[i]+":");
        cal[i]=input.nextInt();
        totcal=totcal+cal[i];
    }
    clrscr1();
    System.out.println("\t\t\t\t\t\t\t\t\t\tSUMMARY\n\n");
    System.out.println("\t\t\t\t\tNAME\t\t\t\t\t\t\t\t\t\tCALORIES\n");
    for(int i=0;i<num;i++)
    {
        System.out.println("\t\t\t\t\t"+name[i]+"\t\t\t\t\t\t\t\t\t\t"+cal[i]+"\n");
    }
    System.out.println("\n\n\n\t\t\t\t\t\t\t\tTOTAL CALORIES CONSUMED = "+totcal);
    if(totcal>=calgoal)
    {
        System.out.println("\n\t\t\t\t\t\t\t\tCONGRATULATIONS!!");
        System.out.println("\n\t\t\t\t\t\t\t\tCALORIE GOAL ACHIEVED\n\n\n");
        System.out.println("\n\n\n\n");
    }
    else
    {
        System.out.println("\n\t\t\t\t\t\t\t\tINSUFFICIENT AMOUNT OF CALORIES CONSUMED");
        System.out.println("\n\t\t\t\t\t\t\t\tCONSUME MORE FOOD ITEMS TO REACH CALORIE GOAL\n\n\n");
        System.out.println("\n\n\n\n");
    }
```

Fig 4.5 Calories consumed method

In this method,the program asks the user for their calorie goal,the food items they have consumed along with the calories of each food item consumed.It then displays the total calories consumed and whether or not the calorie goal is achieved or not.

4.4 MODULE 4 FUNCTIONALITY

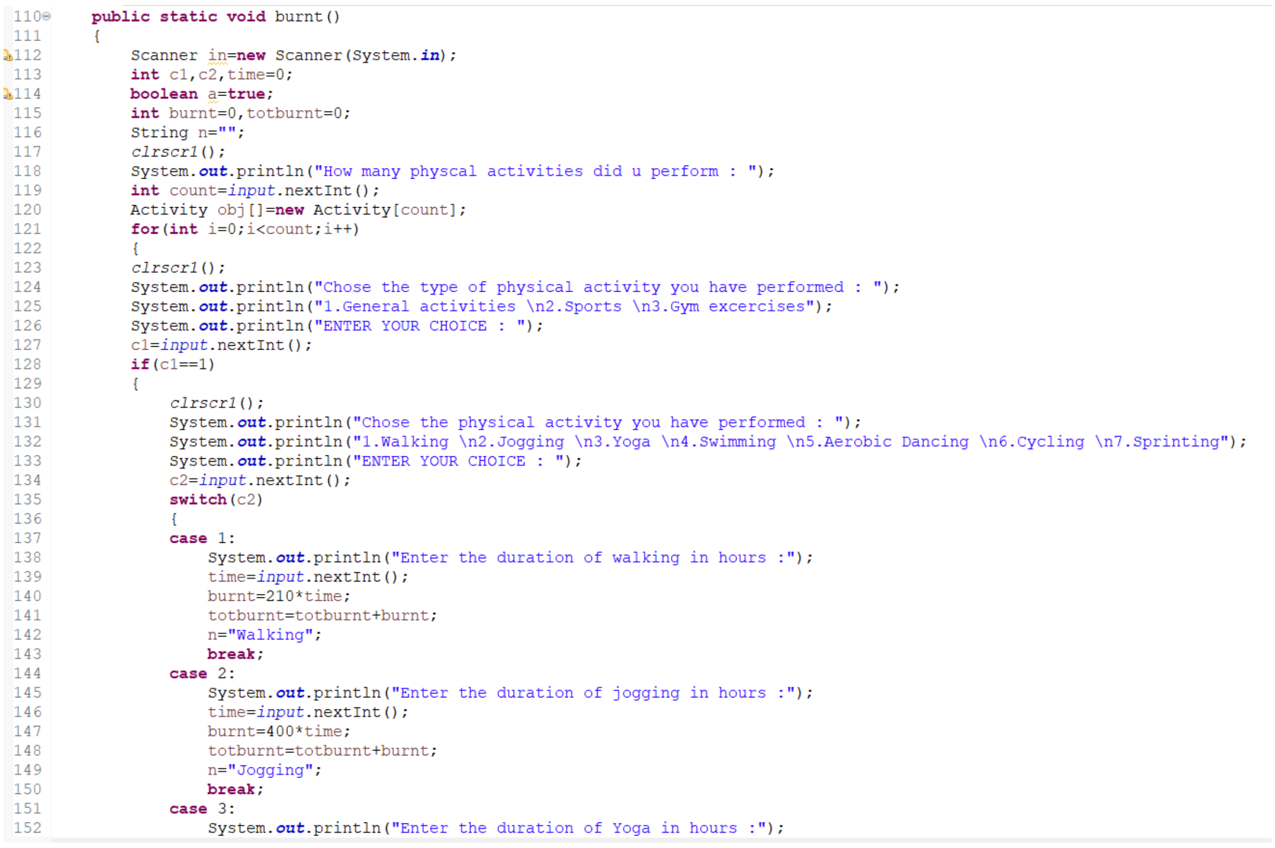


```java
public static void suggestion()
{
    double h,w;
    int a,g=0;
    double main=0,mwg=0,wg=0,fwg=0,mwl=0,wl=0,fwl=0;
    System.out.println("Enter you gender\n\n1.Male\n2.Female\n\nEnter your choice");
    g=input.nextInt();
    System.out.println("\n\n\nEnter your Height in cm :");
    h=input.nextDouble();
    System.out.println("\n\n\nEnter your Weight in kg :");
    w=input.nextDouble();
    System.out.println("\n\n\nEnter your Age :");
    a=input.nextInt();
    h=h/2.54;
    w=w*2.205;
    if(g==1)
    {
        main=66+(6.23*w)+(12.7*h)-(6.8*a);
        main=main*1.2;
    }
    else if(g==2)
    {
        main=655+(4.3*w)+(4.7*h)-(4.7*a);
        main=main*1.2;
    }
    else
    {
        System.out.println("\n\n\nEnter a valid Choice");
    }
    mwg=main+250;
    wg=main+500;
    fwg=main+1000;
    mwl=main-250;
    wl=main-500;
    System.out.println("\n\n\nCalories required to maintaion weight : "+String.format("%.0f",main));
    System.out.println("\n\n\nCalories required for mild weight gain (0.25 KG per week) : "+String.format("%.0f",mwg));
    System.out.println("\n\n\nCalories required for weight gain (0.5 KG per week) : "+String.format("%.0f",wg));
    System.out.println("\n\n\nCalories required for fast weight gain (1 KG per week) : "+String.format("%.0f",fwg));
    System.out.println("\n\n\nCalories required for mild weight loss (0.25 KG per week) : "+String.format("%.0f",mwl));
    System.out.println("\n\n\nCalories required for weight loss (0.5 KG per week) : "+String.format("%.0f",wl));
    System.out.println("\n\n\nCalories required for fast weight loss (1 KG per week) : "+String.format("%.0f",fwl));
    in.nextLine();
}
```

Fig 4.6 Calorie suggestion method

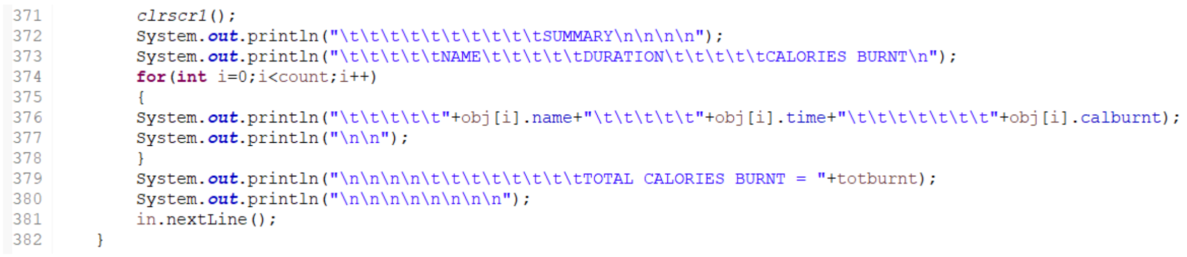
In this method,the program ask the user for their age,gender,height and weight.It then calculates and displays the amount of calories they need to consume for different levels of weight loss or gain.

4.5 MODULE 5 FUNCTIONALITY



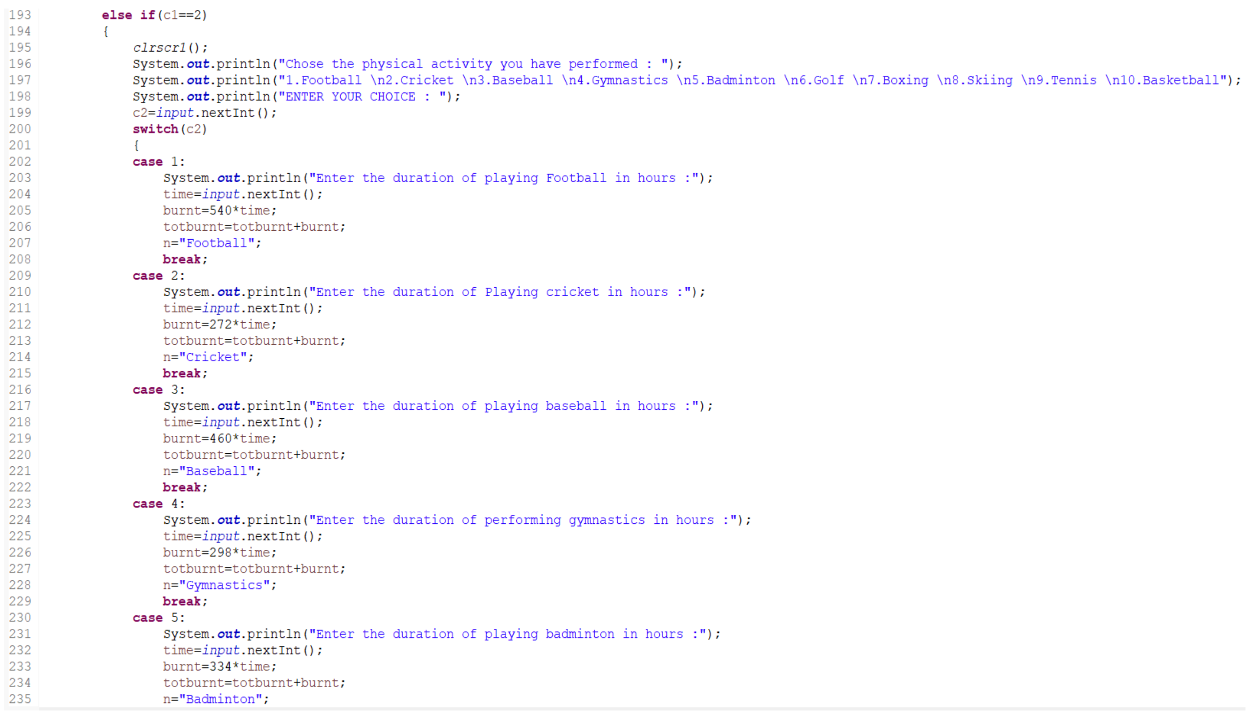
```java
    public static void burnt()
    {
        Scanner in=new Scanner(System.in);
        int c1,c2,time=0;
        boolean a=true;
        int burnt=0,totburnt=0;
        String n="";
        clrscr1();
        System.out.println("How many physcal activities did u perform : ");
        int count=input.nextInt();
        Activity obj[]=new Activity[count];
        for(int i=0;i<count;i++)
        {
        clrscr1();
        System.out.println("Chose the type of physical activity you have performed : ");
        System.out.println("1.General activities \n2.Sports \n3.Gym excercises");
        System.out.println("ENTER YOUR CHOICE : ");
        c1=input.nextInt();
        if(c1==1)
        {
            clrscr1();
            System.out.println("Chose the physical activity you have performed : ");
            System.out.println("1.Walking \n2.Jogging \n3.Yoga \n4.Swimming \n5.Aerobic Dancing \n6.Cycling \n7.Sprinting");
            System.out.println("ENTER YOUR CHOICE : ");
            c2=input.nextInt();
            switch(c2)
            {
            case 1:
                System.out.println("Enter the duration of walking in hours :");
                time=input.nextInt();
                burnt=210*time;
                totburnt=totburnt+burnt;
                n="Walking";
                break;
            case 2:
                System.out.println("Enter the duration of jogging in hours :");
                time=input.nextInt();
                burnt=400*time;
                totburnt=totburnt+burnt;
                n="Jogging";
                break;
            case 3:
                System.out.println("Enter the duration of Yoga in hours :");
```

Fig 4.7 Calories burnt method



```java
        clrscr1();
        System.out.println("\t\t\t\t\t\t\t\t\t\tSUMMARY\n\n\n\n");
        System.out.println("\t\t\t\t\tNAME\t\t\t\t\tDURATION\t\t\t\t\tCALORIES BURNT\n");
        for(int i=0;i<count;i++)
        {
        System.out.println("\t\t\t\t\t"+obj[i].name+"\t\t\t\t\t"+obj[i].time+"\t\t\t\t\t\t\t"+obj[i].calburnt);
        System.out.println("\n\n");
        }
        System.out.println("\n\n\n\n\t\t\t\t\t\t\t\tTOTAL CALORIES BURNT = "+totburnt);
        System.out.println("\n\n\n\n\n\n\n\n");
        in.nextLine();
    }
```

Fig 4.8 Calories burnt method

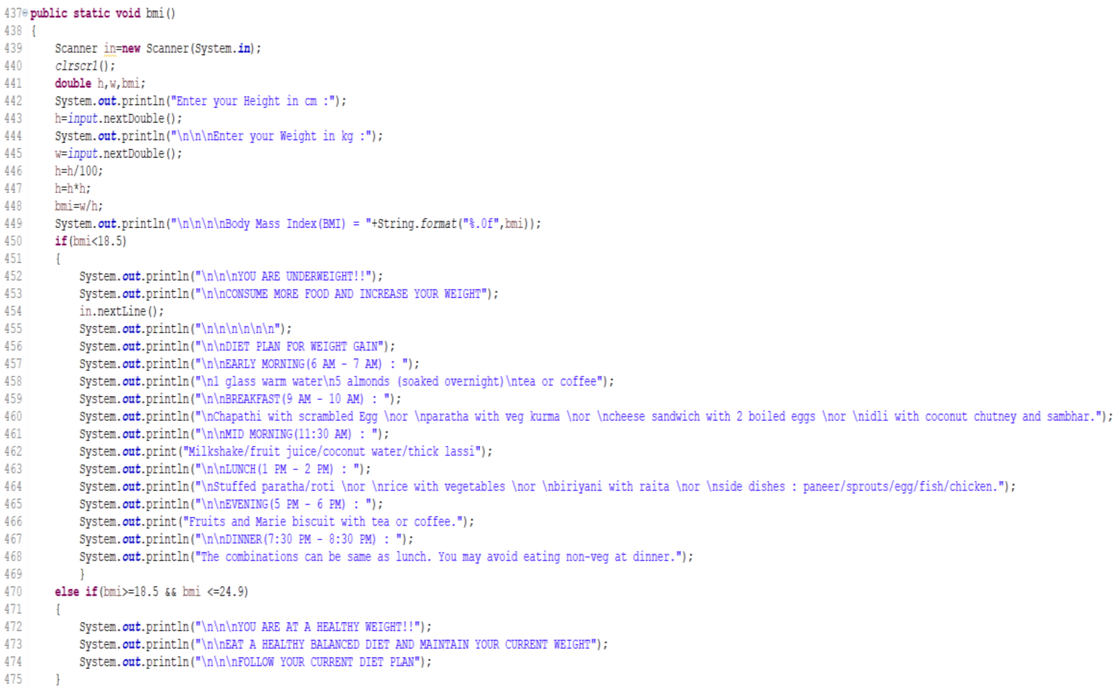


```
        else if(c1==2)
        {
            clrscr1();
            System.out.println("Chose the physical activity you have performed : ");
            System.out.println("1.Football \n2.Cricket \n3.Baseball \n4.Gymnastics \n5.Badminton \n6.Golf \n7.Boxing \n8.Skiing \n9.Tennis \n10.Basketball");
            System.out.println("ENTER YOUR CHOICE : ");
            c2=input.nextInt();
            switch(c2)
            {
            case 1:
                System.out.println("Enter the duration of playing Football in hours :");
                time=input.nextInt();
                burnt=540*time;
                totburnt=totburnt+burnt;
                n="Football";
                break;
            case 2:
                System.out.println("Enter the duration of Playing cricket in hours :");
                time=input.nextInt();
                burnt=272*time;
                totburnt=totburnt+burnt;
                n="Cricket";
                break;
            case 3:
                System.out.println("Enter the duration of playing baseball in hours :");
                time=input.nextInt();
                burnt=460*time;
                totburnt=totburnt+burnt;
                n="Baseball";
                break;
            case 4:
                System.out.println("Enter the duration of performing gymnastics in hours :");
                time=input.nextInt();
                burnt=298*time;
                totburnt=totburnt+burnt;
                n="Gymnastics";
                break;
            case 5:
                System.out.println("Enter the duration of playing badminton in hours :");
                time=input.nextInt();
                burnt=334*time;
                totburnt=totburnt+burnt;
                n="Badminton";
```

Fig 4.9 Calories burnt method

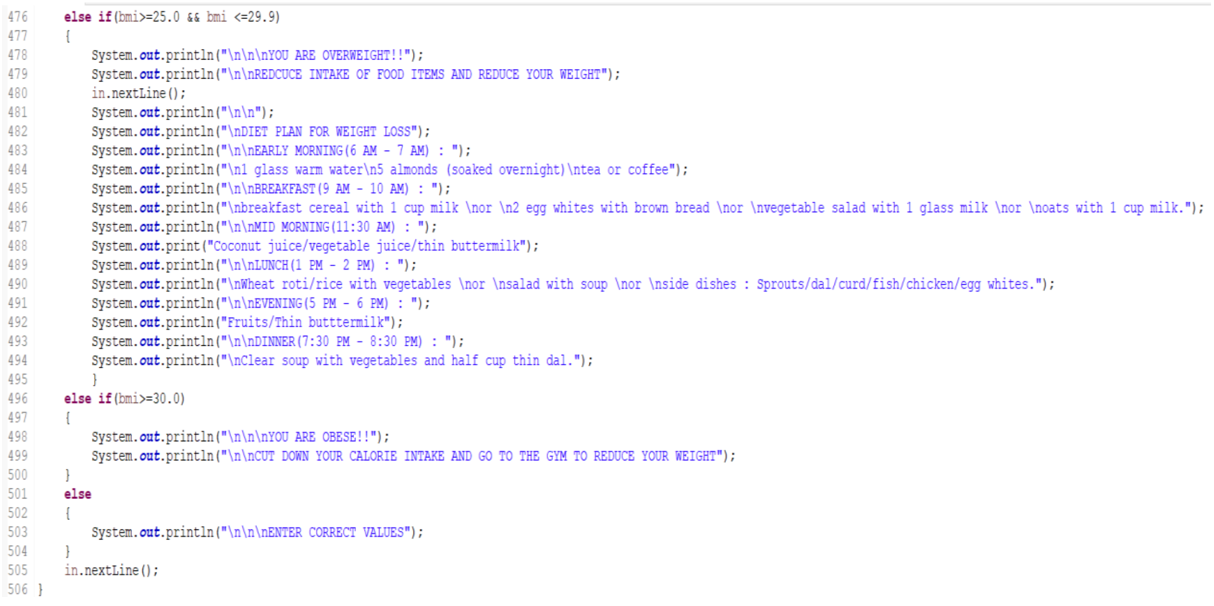
In this method,the program asks the user the different physical activites they have performed along with their respective durations.It then calculates and displays the total amount of calories they have burnt after the ssion of workout.

4.6 MODULE 6 FUNCTIONALITY



```
public static void bmi()
{
    Scanner in=new Scanner(System.in);
    clrscr1();
    double h,w,bmi;
    System.out.println("Enter your Height in cm :");
    h=input.nextDouble();
    System.out.println("\n\n\nEnter your Weight in kg :");
    w=input.nextDouble();
    h=h/100;
    h=h*h;
    bmi=w/h;
    System.out.println("\n\n\n\nBody Mass Index(BMI) = "+String.format("%.0f",bmi));
    if(bmi<18.5)
    {
        System.out.println("\n\n\nYOU ARE UNDERWEIGHT!!");
        System.out.println("\n\nCONSUME MORE FOOD AND INCREASE YOUR WEIGHT");
        in.nextLine();
        System.out.println("\n\n\n\n\n\n");
        System.out.println("\n\nDIET PLAN FOR WEIGHT GAIN");
        System.out.println("\n\nEARLY MORNING(6 AM - 7 AM) : ");
        System.out.println("\n1 glass warm water\n5 almonds (soaked overnight)\ntea or coffee");
        System.out.println("\n\nBREAKFAST(9 AM - 10 AM) : ");
        System.out.println("\nChapathi with scrambled Egg \nor \nparatha with veg kurma \nor \ncheese sandwich with 2 boiled eggs \nor \nidli with coconut chutney and sambhar.");
        System.out.println("\n\nMID MORNING(11:30 AM) : ");
        System.out.print("Milkshake/fruit juice/coconut water/thick lassi");
        System.out.println("\n\nLUNCH(1 PM - 2 PM) : ");
        System.out.println("\nStuffed paratha/roti \nor \nrice with vegetables \nor \nbiriyani with raita \nor \nside dishes : paneer/sprouts/egg/fish/chicken.");
        System.out.println("\n\nEVENING(5 PM - 6 PM) : ");
        System.out.print("Fruits and Marie biscuit with tea or coffee.");
        System.out.println("\n\nDINNER(7:30 PM - 8:30 PM) : ");
        System.out.println("The combinations can be same as lunch. You may avoid eating non-veg at dinner.");
        }
    else if(bmi>=18.5 && bmi <=24.9)
    {
        System.out.println("\n\n\nYOU ARE AT A HEALTHY WEIGHT!!");
        System.out.println("\n\nEAT A HEALTHY BALANCED DIET AND MAINTAIN YOUR CURRENT WEIGHT");
        System.out.println("\n\n\nFOLLOW YOUR CURRENT DIET PLAN");
    }
```

Fig 4.10 BMI method

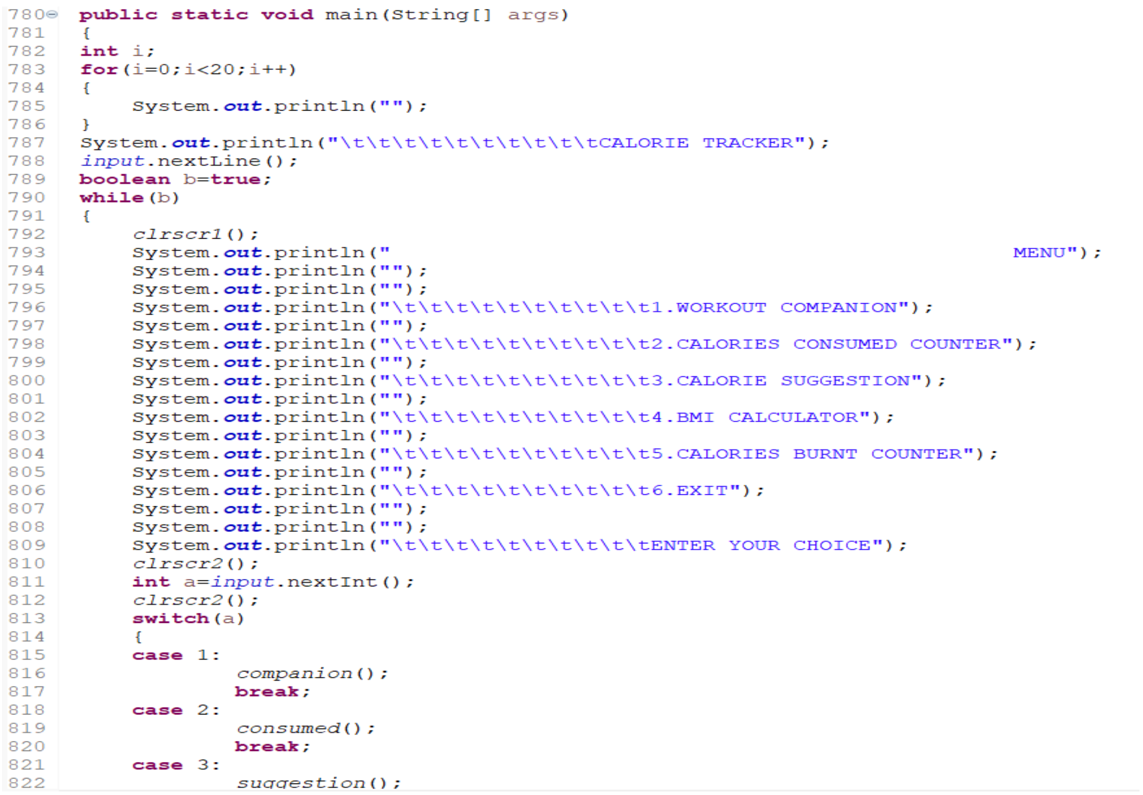


```java
    else if(bmi>=25.0 && bmi <=29.9)
    {
        System.out.println("\n\n\nYOU ARE OVERWEIGHT!!");
        System.out.println("\n\nREDCUCE INTAKE OF FOOD ITEMS AND REDUCE YOUR WEIGHT");
        in.nextLine();
        System.out.println("\n\n");
        System.out.println("\nDIET PLAN FOR WEIGHT LOSS");
        System.out.println("\n\nEARLY MORNING(6 AM - 7 AM) : ");
        System.out.println("\n1 glass warm water\n5 almonds (soaked overnight)\ntea or coffee");
        System.out.println("\n\nBREAKFAST(9 AM - 10 AM) : ");
        System.out.println("\nbreakfast cereal with 1 cup milk \nor \n2 egg whites with brown bread \nor \nvegetable salad with 1 glass milk \nor \noats with 1 cup milk.");
        System.out.println("\n\nMID MORNING(11:30 AM) : ");
        System.out.print("Coconut juice/vegetable juice/thin buttermilk");
        System.out.println("\n\nLUNCH(1 PM - 2 PM) : ");
        System.out.println("\nWheat roti/rice with vegetables \nor \nsalad with soup \nor \nside dishes : Sprouts/dal/curd/fish/chicken/egg whites.");
        System.out.println("\n\nEVENING(5 PM - 6 PM) : ");
        System.out.println("Fruits/Thin butttermilk");
        System.out.println("\n\nDINNER(7:30 PM - 8:30 PM) : ");
        System.out.println("\nClear soup with vegetables and half cup thin dal.");
        }
    else if(bmi>=30.0)
    {
        System.out.println("\n\n\nYOU ARE OBESE!!");
        System.out.println("\n\nCUT DOWN YOUR CALORIE INTAKE AND GO TO THE GYM TO REDUCE YOUR WEIGHT");
    }
    else
    {
        System.out.println("\n\n\nENTER CORRECT VALUES");
    }
    in.nextLine();
}
```

Fig 4.11 BMI method

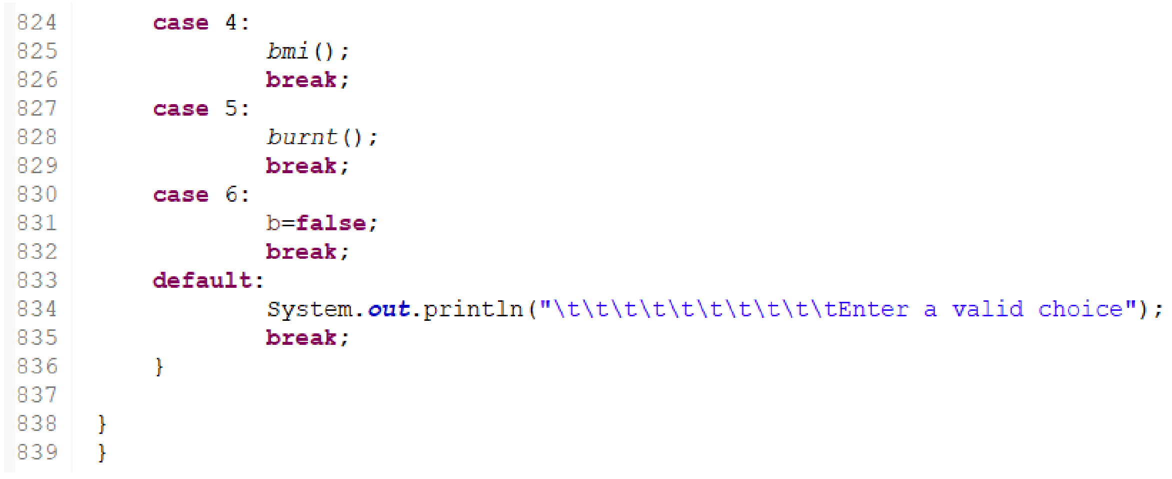
I this method,the program asks the user for their height and weight.It then calculates and displays the user's BMI.It also displays a customised diet plan for the user according to their BMI value.

4.7 MODULE 7 FUNCTIONALITY



```java
public static void main(String[] args)
{
int i;
for(i=0;i<20;i++)
{
    System.out.println("");
}
System.out.println("\t\t\t\t\t\t\t\t\t\tCALORIE TRACKER");
input.nextLine();
boolean b=true;
while(b)
{
    clrscr1();
    System.out.println("                                                              MENU");
    System.out.println("");
    System.out.println("");
    System.out.println("\t\t\t\t\t\t\t\t\t\t1.WORKOUT COMPANION");
    System.out.println("");
    System.out.println("\t\t\t\t\t\t\t\t\t\t2.CALORIES CONSUMED COUNTER");
    System.out.println("");
    System.out.println("\t\t\t\t\t\t\t\t\t\t3.CALORIE SUGGESTION");
    System.out.println("");
    System.out.println("\t\t\t\t\t\t\t\t\t\t4.BMI CALCULATOR");
    System.out.println("");
    System.out.println("\t\t\t\t\t\t\t\t\t\t5.CALORIES BURNT COUNTER");
    System.out.println("");
    System.out.println("\t\t\t\t\t\t\t\t\t\t6.EXIT");
    System.out.println("");
    System.out.println("");
    System.out.println("\t\t\t\t\t\t\t\t\t\tENTER YOUR CHOICE");
    clrscr2();
    int a=input.nextInt();
    clrscr2();
    switch(a)
    {
    case 1:
            companion();
            break;
    case 2:
            consumed();
            break;
    case 3:
            suggestion();
```

Fig 4.12 Main method



```
        case 4:
                bmi();
                break;
        case 5:
                burnt();
                break;
        case 6:
                b=false;
                break;
        default:
                System.out.println("\t\t\t\t\t\t\t\t\t\tEnter a valid choice");
                break;
        }

    }
    }
```

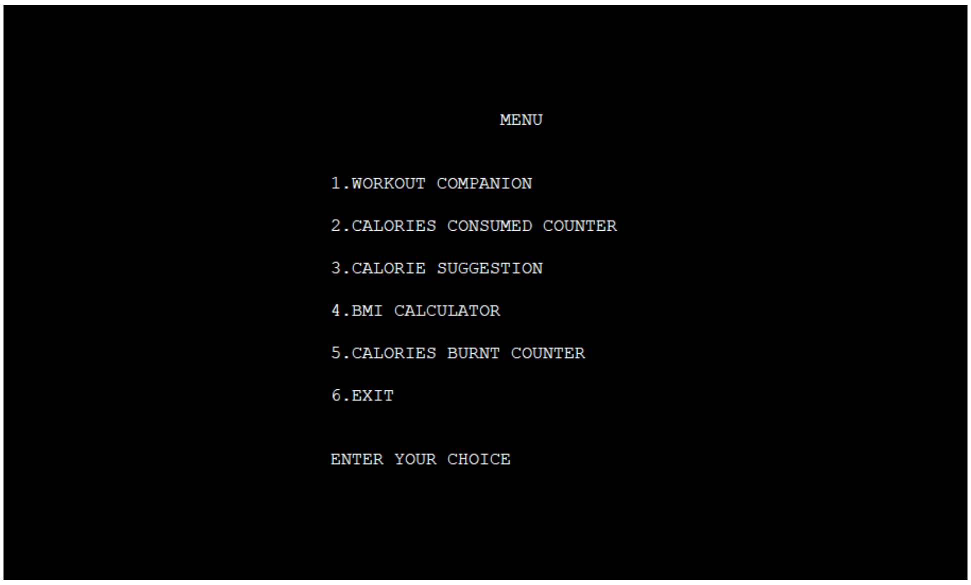
Fig 4.13 Main method

This is the main method where the program execution begins.This method is used to display the main menu and it calls the other methods according to the input given by the user.

CHAPTER 5

RESULTS

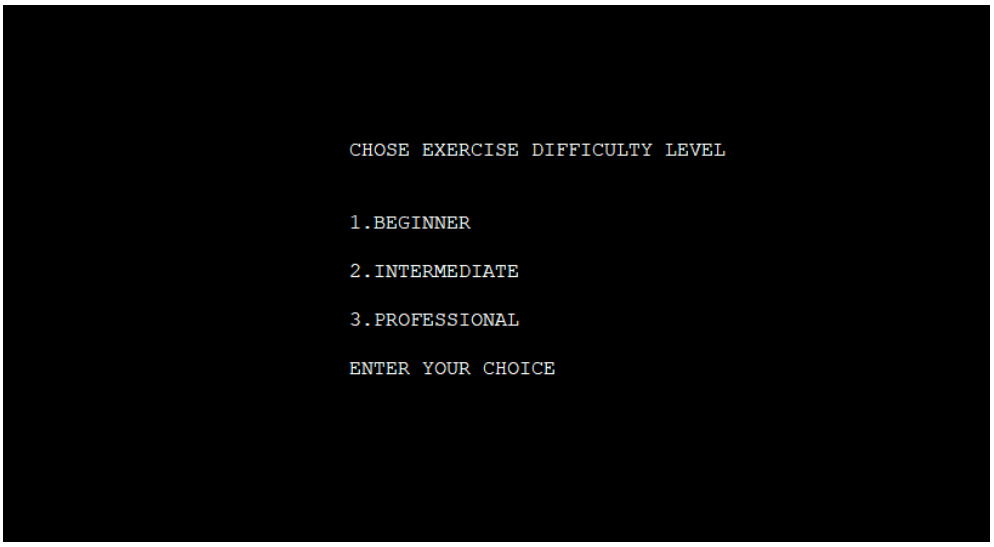
5.1 Snapshots



```
                                   MENU

                  1.WORKOUT COMPANION

                  2.CALORIES CONSUMED COUNTER

                  3.CALORIE SUGGESTION

                  4.BMI CALCULATOR

                  5.CALORIES BURNT COUNTER

                  6.EXIT

                  ENTER YOUR CHOICE
```

Fig 5.1 Main Menu

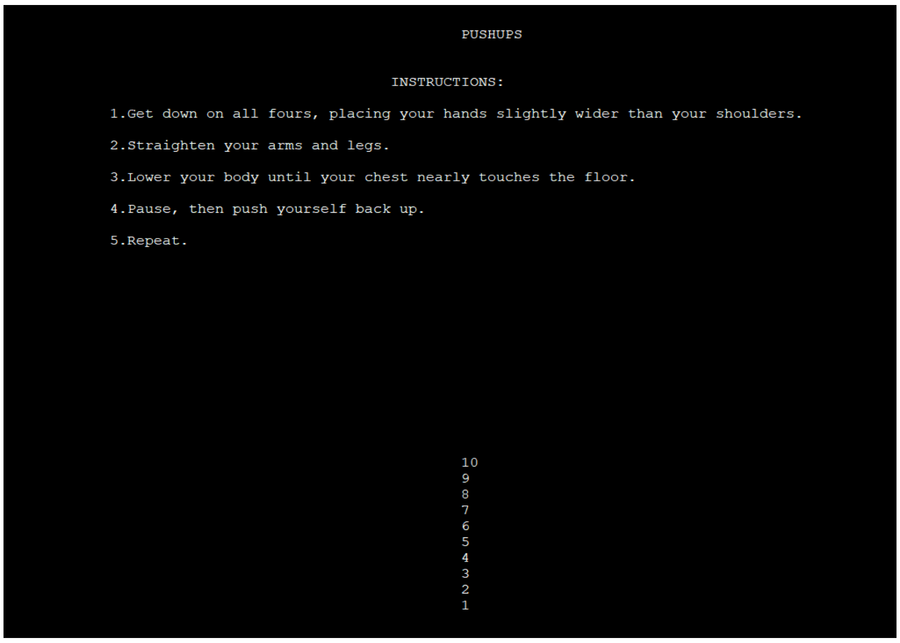
Main menu of the program.



```
CHOSE EXERCISE DIFFICULTY LEVEL

1.BEGINNER

2.INTERMEDIATE

3.PROFESSIONAL

ENTER YOUR CHOICE
```

Fig 5.2 Workout Companion

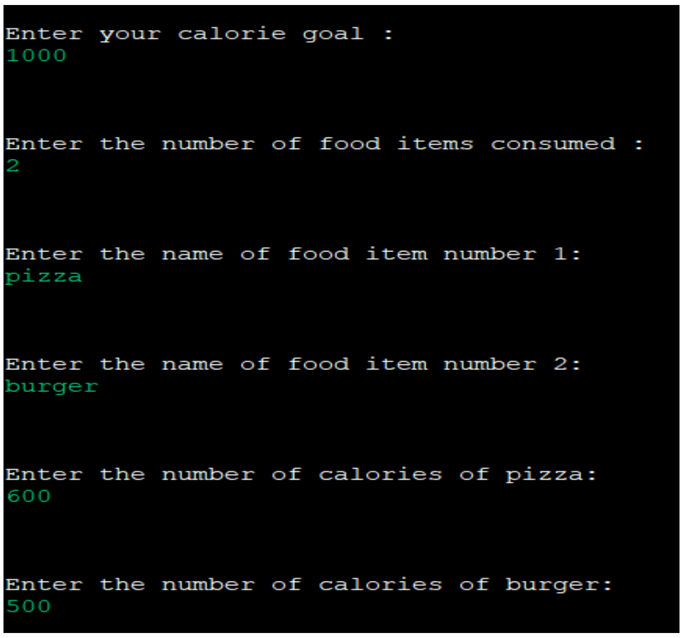
This is the workout companion method in which there are 3 options beginner,intermediate,professional.



```
                         PUSHUPS

                      INSTRUCTIONS:
1.Get down on all fours, placing your hands slightly wider than your shoulders.
2.Straighten your arms and legs.
3.Lower your body until your chest nearly touches the floor.
4.Pause, then push yourself back up.
5.Repeat.

                         10
                         9
                         8
                         7
                         6
                         5
                         4
                         3
                         2
                         1
```

Fig 5.3 Workout Companion

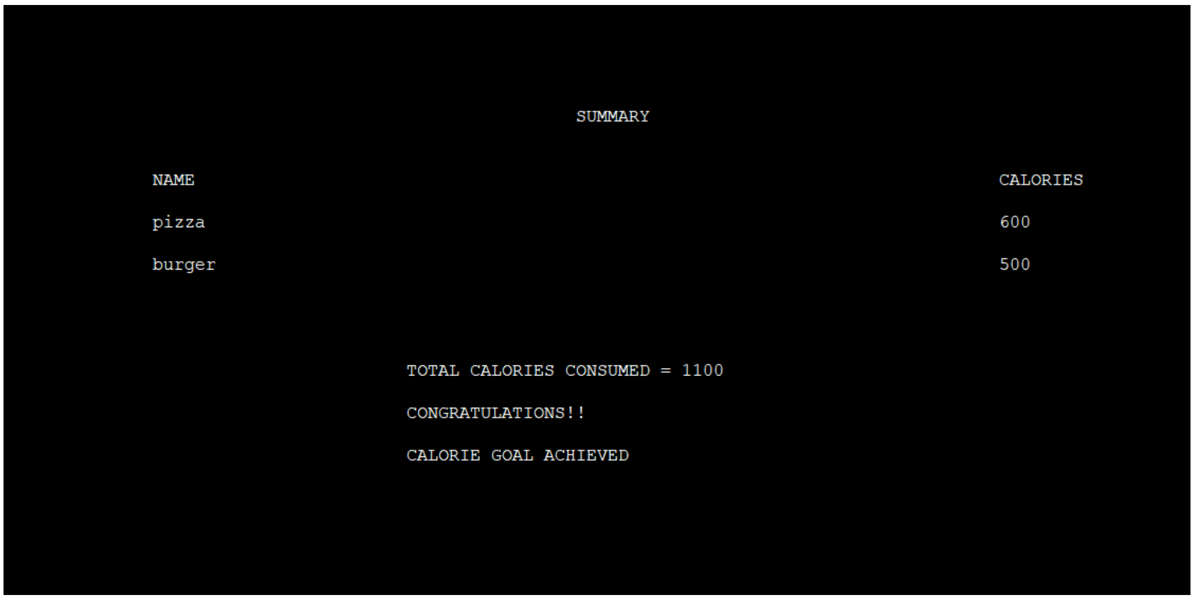
Workout companion in which the timer of 10 seconds has been displayed for one of the beginner level exercises that is push ups.



```
Enter your calorie goal :
1000


Enter the number of food items consumed :
2


Enter the name of food item number 1:
pizza


Enter the name of food item number 2:
burger


Enter the number of calories of pizza:
600


Enter the number of calories of burger:
500
```

Fig 5.4 Calories consumed

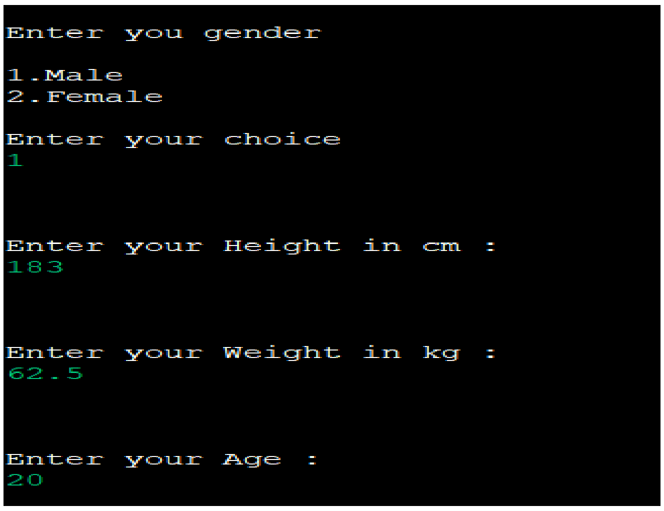
This is the calories consumed counter in which the input for 2 food items pizza and burger for 600 and 500 calories respectively are entered.



```
                              SUMMARY

NAME                                                        CALORIES

pizza                                                       600

burger                                                      500


                TOTAL CALORIES CONSUMED = 1100

                CONGRATULATIONS!!

                CALORIE GOAL ACHIEVED
```

Fig 5.5 Calories consumed

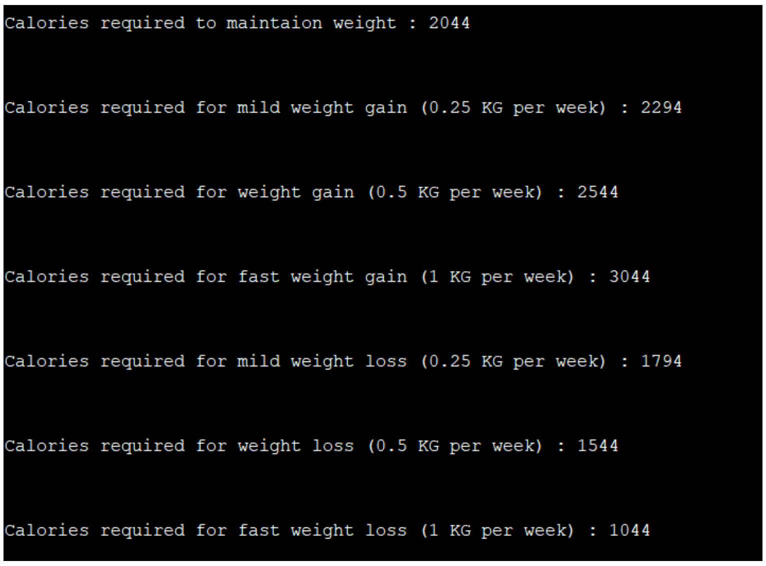
Summary of the food items consumed and the total amount calories consumed along with the display whether the calorie goal is achieved or not.



```
Enter you gender

1.Male
2.Female

Enter your choice
1


Enter your Height in cm :
183


Enter your Weight in kg :
62.5


Enter your Age :
20
```

Fig 5.6 Calorie suggestion

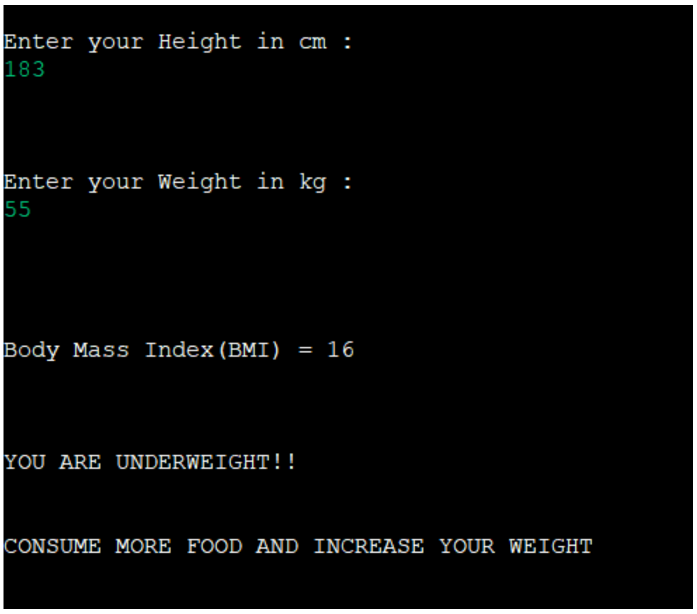
This is the calorie suggestion method in which the input for gender,height,weight and age is entered.



```
Calories required to maintaion weight : 2044

Calories required for mild weight gain (0.25 KG per week) : 2294

Calories required for weight gain (0.5 KG per week) : 2544

Calories required for fast weight gain (1 KG per week) : 3044

Calories required for mild weight loss (0.25 KG per week) : 1794

Calories required for weight loss (0.5 KG per week) : 1544

Calories required for fast weight loss (1 KG per week) : 1044
```

Fig 5.7 Calorie suggestion

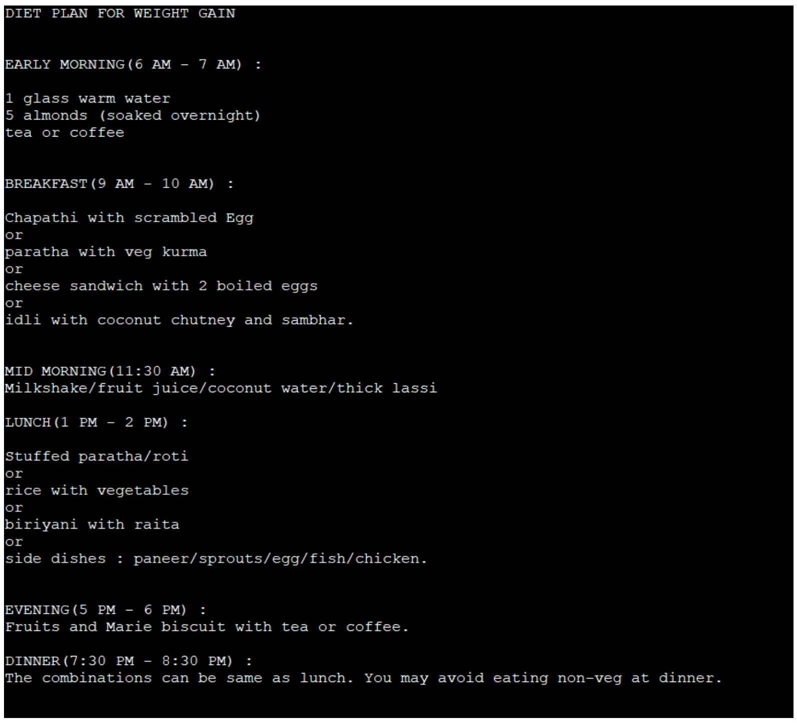
Calorie suggestion method in which the various calorie requirements according to the given input to achieve different levels of weight loss or weight gain are displayed.



```
Enter your Height in cm :
183


Enter your Weight in kg :
55



Body Mass Index(BMI) = 16


YOU ARE UNDERWEIGHT!!

CONSUME MORE FOOD AND INCREASE YOUR WEIGHT
```

Fig 5.8 BMI Calculator

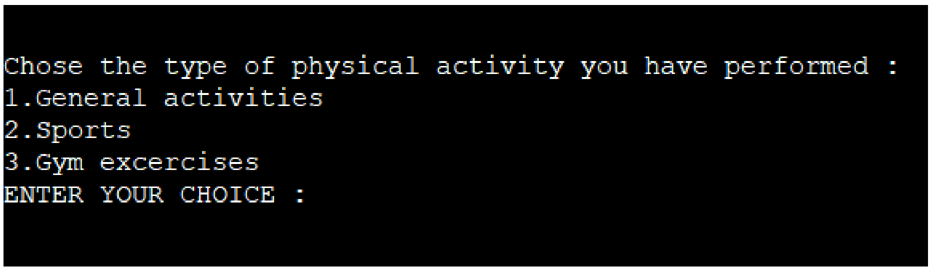
This is the BMI calculator method in which the input for height and weight are given.The output of the BMI is displayed here.



```
DIET PLAN FOR WEIGHT GAIN


EARLY MORNING(6 AM - 7 AM) :

1 glass warm water
5 almonds (soaked overnight)
tea or coffee


BREAKFAST(9 AM - 10 AM) :

Chapathi with scrambled Egg
or
paratha with veg kurma
or
cheese sandwich with 2 boiled eggs
or
idli with coconut chutney and sambhar.


MID MORNING(11:30 AM) :
Milkshake/fruit juice/coconut water/thick lassi

LUNCH(1 PM - 2 PM) :

Stuffed paratha/roti
or
rice with vegetables
or
biriyani with raita
or
side dishes : paneer/sprouts/egg/fish/chicken.


EVENING(5 PM - 6 PM) :
Fruits and Marie biscuit with tea or coffee.

DINNER(7:30 PM - 8:30 PM) :
The combinations can be same as lunch. You may avoid eating non-veg at dinner.
```

Fig 5.9 BMI Calculator

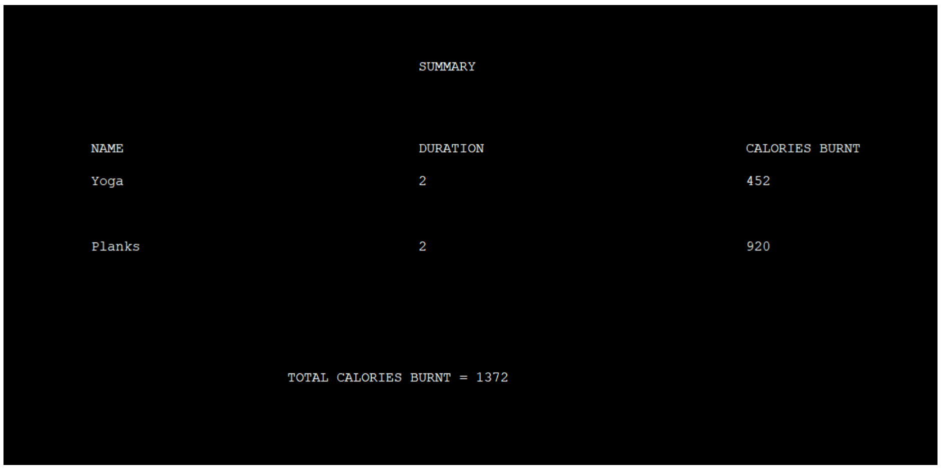
BMI calculator method showing the customised diet plan for the user who is underweight.



```
Chose the type of physical activity you have performed :
1.General activities
2.Sports
3.Gym excercises
ENTER YOUR CHOICE :
```

Fig 5.10 Calories burnt

This is the calorie burnt counter which gives the option to select between different types of physical activity.



```
                                        SUMMARY


NAME                                    DURATION                                CALORIES BURNT
Yoga                                    2                                       452


Planks                                  2                                       920




                                TOTAL CALORIES BURNT = 1372
```

Fig 5.11 Calories burnt

Calorie burnt counter displaying the summary of all the physical activities performed along with duration of each activity.It also displays the total amount of calories burnt after the session.

CHAPTER 6

CONCLUSION

The mini project has successfully accomplished the goals it had set out in the objectives and design sections of this report.

This mini project serves as an all round health companion to the users as it helps is simplifying the process of keeping track of the amount of calories consumed and burnt.It also allows the users to set up timed workout sessions of different difficulty levels.This project also gives the users detailed information about the calories they need to maintain on a daily basis if they need to gain or lose weight.It also makes the process of calculating the BMI very easy and it provides the user with customised diet plans according to their BMI value.

This mini project provides an user friendly interface for the users so that they can keep track of their fitness and help them in their weight loss/gain journey.

REFERENCES

1) https://www.javatpoint.com

2) https://www.programiz.com

3) https://www.geeksforgeeks.org

4) https://www.tutorialspoint.com