RIT | Golisano College of Computing and Information Sciences
School of Information

ISTE-120 - Computational Problem Solving for the Information Domain I
Homework Assignment 7 (HW07)

"How Healthy Are You?"

Problem

Write a program that calculates health information for people that computes both the Basal Metabolic Rate (BMR) and the Body Mass Index (BMI). BMR and BMI are calculated for a specific person.

Programming focus

This program uses conditional statements, if, if-else-if, and switch.

Description

Prompt the user to enter his/her name, age, gender, weight in pounds, height in inches, and activity level. The program must validate this input based on the following criteria:

Input	Rules to Validate Input
Name	At least one character
Gender	M or F (upper or lower case
Weight	At least 100 pounds
Height	Between 60 to 84 inches, inclusively
Age	At least 18 years old
Activity Level	Between 1 to 5, inclusively

Based on valid input, the program must calculate the user's BMR, BMI, and print some other useful information for the user.

Equations

The Harris-Benedict formula computes the BMR based on total body weight. The formula is based on the Metric system (kilograms and centimeters) that is calculated as follows:

Men: BMR = 66 + (13.7 * wt in kg) + (5 * ht in cm) - (6.8 * age in years)
Women: BMR = 655 + (9.6 * wt in kg) + (1.8 * ht in cm) - (4.7 * age in years)

Using accurate conversion factors, convert from inches and pounds to the metric values for use in the above equations. Research these conversion formulas on the internet.

The calculation for the BMI is in the English measurements system (pounds and inches) and is given as follows:

BMI = ((weight in pounds) / (height in inches)^2) * 703

The following ranges of the BMI give the user an indication of the overall status of the user's weight:

- Underweight if BMI < 18.5
- Normal weight if BMI >= 18.5 and < 25
- Overweight if BMI >= 25 and < 30
- Obese if BMI >= 30

The user's activity level is used to calculate the user's Total Daily Energy Expenditure (TDEE) in calories. The user's TDEE is calculated by multiplying the user's BMR by the user's activity multiplier from the chart below:

Activity Level	Activity Multiplier	Description
Sedentary	BMR * 1.2	(little or no exercise, desk job)
Lightly active	BMR * 1.375	(light exercise/sports 1-3 days/wk)
Moderately active	BMR * 1.55	(moderate exercise/sports 3-5 days/wk)
Very active	BMR * 1.725	(hard exercise/sports 6-7 days/wk)
Extra active	BMR * 1.9	(hard daily exercise/sports & physical job or 2 * day training, i.e. marathon, contest etc.)

Programming Requirements:

1. The main method in the class **HowHealthy** accepts all of the user input, validates all of the user input, and prints all the results

2. The main method in the class **HowHealthy** prompts the user for the following in the order given: name, gender, weight in pounds, height in inches, age in years, and Activity Level

3. The output contains the input values, the BMR, the BMI, the TDEE and the overall status of the user's weight based on the range analysis of the BMI. Print the BMR, BMI and TDEE with two decimal places. Use the appropriate accessor to print the value of each attribute

4. The functional class, **Healthy,** must contain attributes required for all calculations

5. The class **Healthy** must have a constructor to initialize all of the attributes

6. The class **Healthy** must do all of the calculations and must **NOT** have any input or output. In other words, the Scanner class and System.out.println/print methods cannot be used in the Healthy class

7. The class **Healthy** must have an accessor for each attribute

8. The class **Healthy** must include **private** methods to do the metric-to-English conversions. These are **private** because they are only used in Healthy and are not to be used by any other class

Programming Notes:

1. To call a method within the same class, call the method without using any object name

2. The values calculated by the instructor's solution varies slightly from the values calculated on the given website. The differences seem to stem from conversion factors from Imperial to Metric

3. Use the web to find accurate conversion factors. In most cases, 1 or 2 decimal places is NOT sufficient

Program Design:

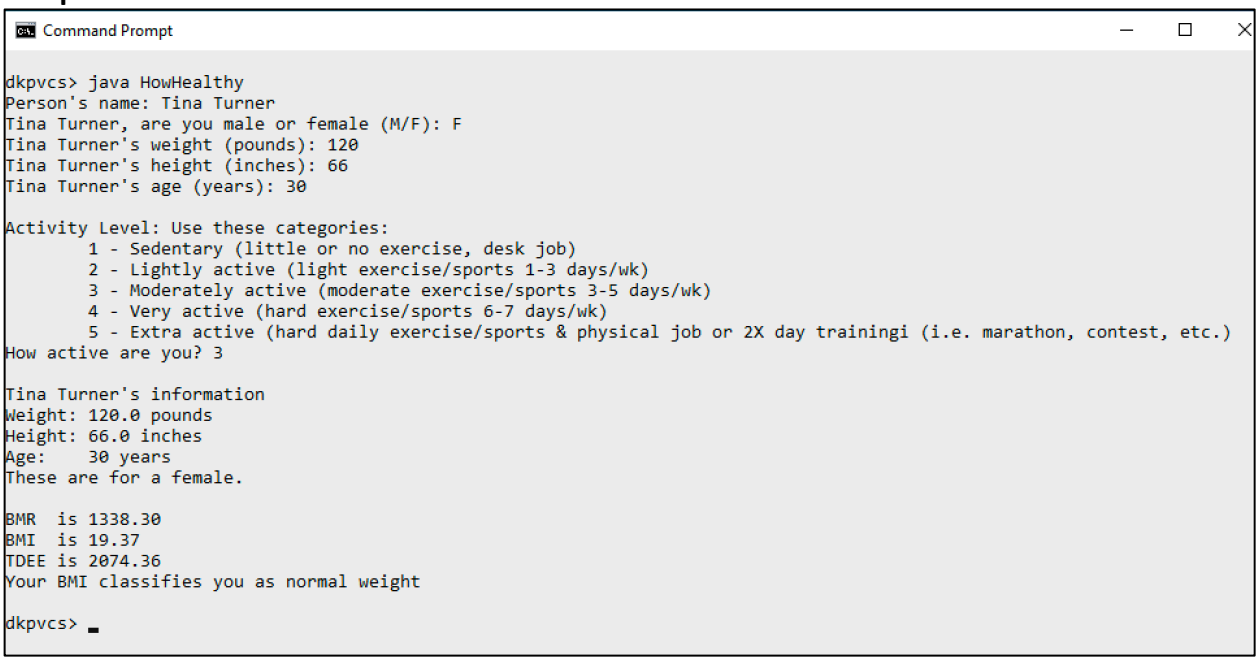
Before writing code, carefully design the program using a few ideas discussed below.

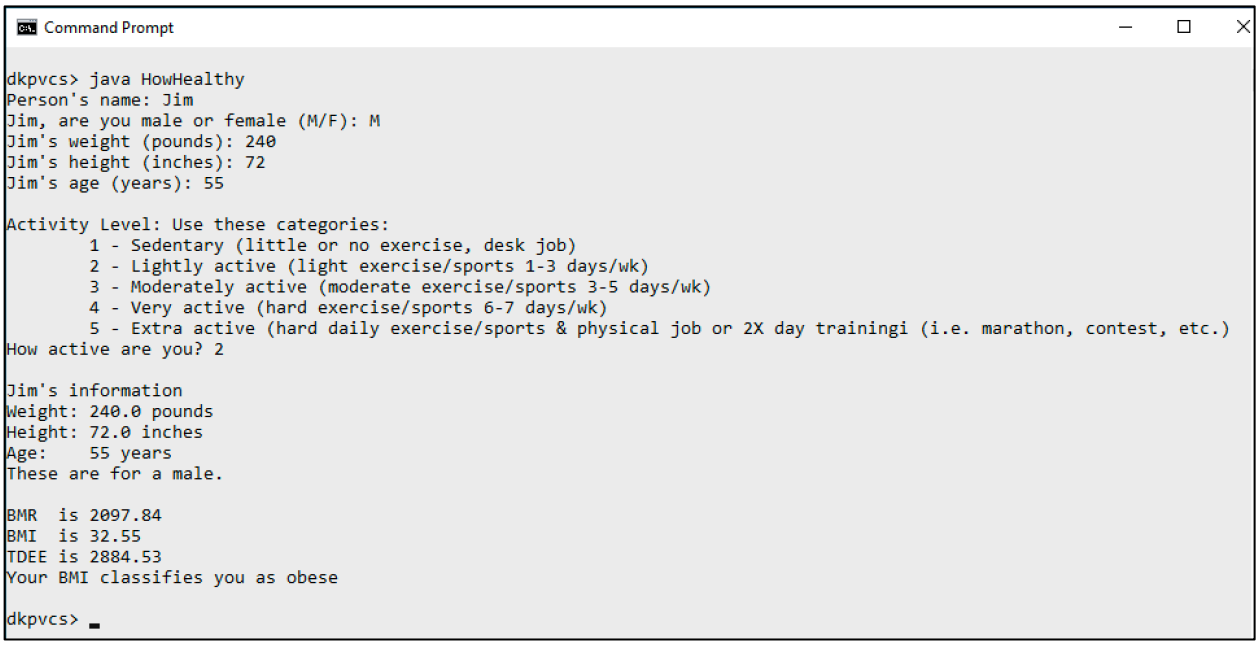
For the main method, determine the actions that need to be carried out. One common technique is to highlight or underline phrases in the program description. Then order the actions so that they are carried out in the proper sequence

For the functional class, identify the data items the class requires. Then identify the actions that the class needs to carry out

Do not delay beginning this homework. Allow time for questions to your instructor.

Sample Executions





SUBMISSION

Submit a zip file that contains the two java files to the Homework 7 dropbox on myCourses. See the dropbox for submission deadline.

Homework 7 Grade Sheet

Program Design/Requirements	Point Value	Points Earned
HowHealthy.java		
• Reads all of input	10	
• Validates all the input and prints appropriate error messages for invalid input	10	
• Prints all of the output using appropriate accessor.	10	
• Calls methods of the Healthy class to do all calculations	10	
Healthy.java		
• Contains all required attributes.	2	
• Contains constructor to initialize all attributes.	5	
• Contains method to calculate BMR.	5	
• Contains a method to calculate BMI.	5	
• Contains a method to calculate TDEE.	5	
• Contains a method to determine weight classification.	5	
• Contains private methods to do imperial-metric conversion.	5	
• "Special" numbers are kept as constants.	3	
Overall:		
• Output matches sample shown	25	
Total Points	100	

Coding Standards: (Points deducted after Program Design/Requirements Points are assigned.)

- Naming Conventions: meaningful; use of capitalization and underscores appropriate for class, method, variables, and constants
  Deduction (0-10 pts) ____

- Code comments: Header for each file; methods, major logic section as needed
  Deduction (0-10 pts) ____

- Alignment & Indentation: This includes class, methods, ifs, switch, and others.
  Deduction (0-10 pts) ____

Coding Standards Deductions: _____

Grade (Total Points Earned – Coding Standards Deductions): _____