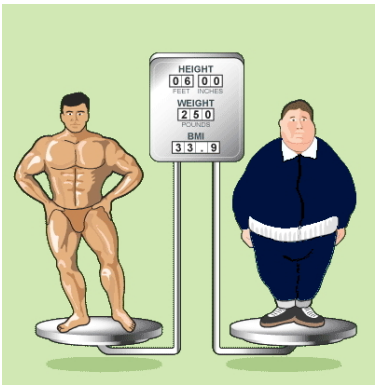
Mr. Cochran
AP Computer Science

Lab 02.01: BMI Calculator
(see Veracross for a due date/time)

Lab Description:



Write a class called `BMICalculator` that prompts the user with two input statements:

1. Enter your height in feet and inches (format 6'2"):
2. Enter your weight in pounds:

The computer must wait for the user to respond to each question. Following the input data from the user, the computer will calculate the user's BMI, or body mass index, handled by the following formula.

$$BMI = \frac{weight(kg)}{height(m)^2}$$

Note: The user is asked to supply their height in feet and inches measurement in the format F'I" where F represents feet and I represents inches. **Your program does not need to handle invalid entry format other than the user entering a value of 0 for their height;** instead, you should plan to use the `String` methods `indexOf` and `substring` to extract the digit values from the user input.

Important Conversion and Output Information

1 inch == 0.0254 meters
1 pound == 0.454 kg
NOTE you will have to convert the supplied information from the user into the appropriate metric values.

For output conversion, you will be required to use a `DecimalFormat` object (see a link to a tutorial on the Unit 2 course page) so that the user's BMI is output to the screen with a maximum of 2 decimal places (including trailing zeros, if they exist).

Sample input/output:

```
Enter your height in feet and inches (Ex 6'1"): 5'11" (user entry)
Enter your weight in pounds: 175 (user entry)
Your BMI, expressed as weight(kg)/height(m)^2: 24.43 kg/m^2
```

Required Methods and `private` Data:

Your version of `BMICalculator` must contain the following methods:

```
/** Convert English to metric units, perform the BMI calculation.
 *  NOTE: this method must properly handle bad data */
public static double computeBMI(int inches, int pounds)

/** Uses a Scanner to prompt the user for info, process the
 *  feet/inches conversion, calls the computeBMI method and prints the
 *  correct information. */
public static void main(String[] args)
```

Documentation Requirements:

Your version of `BMICalculator` must include the following comments/documentation:

A standard 3-line Javadoc header comment
A Javadoc comment for each of the two methods of the class

Testing specifications:

Your `BMICalculator.java` file will be run against a JUnit test called `BMICalcTester.java`. `BMICalcTester.java` will only test your `public static double computeBMI(int inches, int pounds)` method and it must pass all of the tests to receive credit. The `BMICalcTester.java` file will be linked to this lab description.

What to hand in:

Please hand in the following file through the Veracross drop box:

`BMICalculator.java`

DO NOT submit a `.class` file!

Grading rubric:

Description	Points
`BMICalculator.java` contains correct 3-line Javadoc header comment	3
`BMICalculator.java` contains correct Javadoc method documentation	2
`BMICalculator.java` compiles without errors	4
`BMICalculator.java` executes to completion without crashing	4
`BMICalculator.java` correctly utilizes the `String` methods `indexOf` and `substring` as well as the `Integer.parseInt` method to parse user input into useful data for BMI calculation.	3
`BMICalculator.java` correctly utilizes a `DecimalFormat` object to format the output to two decimal places for BMI calculation.	2
`BMICalculator.java` produces the exact output/input as specified in the lab description.	3
`BMICalculator.java` passes all 16 of the supplied JUnit test assertions	8
Code Coverage Rubric At least 50% code coverage At least 60% code coverage At least 70% code coverage At least 80% code coverage At least 90% code coverage	 +2 points +4 points +6 points +8 points +10 points
Student submits the correct file that is properly named to the Veracross drop box, meeting the lab submission deadline	6
TOTAL POINTS available for Lab 02.01 BMI Calculator	**45 points**

Honor Code policy:

You may ask a classmate for help with your code and any associated algorithms, but you may not directly share your code with a classmate. Sharing code in any manner (email, texting, printed copies, etc.) as well as precise and exact copying of a classmate's code is considered a clear-cut violation of Durham Academy's Honor Code. You may also use code or algorithm assistance found online. If you ask someone for help or look at a program online, you must list the names of your classmate(s) with whom you worked or put the URL of the example project in the header comment of your program. All programs will potentially be examined for copied code using the Unix `diff` utility program. If your code is very close to that of a classmate, at a minimum you will have a discussion with your teacher. *Most violations of this policy in years past have been sent to the school's Honor Council for additional investigation.*