

Using Data in a TextArea

In the Body Index Calculator you will be introduced to the following new concepts:

- New package java.io
- Declare and Constructing variables
- Math methods and operations Math.pow(), Math.round() "/" division and "+" plus
- BufferedReader, InputStreamReader
- parseInt(), ReadLine()

Review the application below. Look at each section and in vision what the code will do.

```
1  /*
2  Chapter 2: The Body Mass Index Calculator
3  Programmer: Wendy Plourde
4  Date: October 20, 2008
5  Filename: BodyMass.java
6  Purpose: This project calculates the body mass index based
7           on a person's height and weight.
8  */
9
10 import java.io.*;
11
12 public class BodyMass
13 {
14     public static void main(String[] args) throws IOException //Used with input and output
15     {
16         // declare and construct variables
17         String height, weight;
18         int inches, pounds;
19         double kilograms, meters, index;
20         BufferedReader dataIn= new BufferedReader(new InputStreamReader(System.in));
21
22         // print prompts and get input
23         System.out.println("\tTHE SUN FITNESS CENTER BODY MASS INDEX CALCULATOR");
24         System.out.println();
25         System.out.print("\t\tEnter your height to the nearest inch: ");
26         height = dataIn.readLine();
27         inches = Integer.parseInt(height);
28         System.out.print("\t\tEnter your weight to the nearest pound: ");
29         weight = dataIn.readLine();
30         pounds = Integer.parseInt(weight);
31
32         // calculations
33         meters = inches / 39.36;
34         kilograms = pounds / 2.2;
35         index = kilograms / Math.pow(meters,2);
36
37         // output
38         System.out.println();
39         System.out.println("\tYOUR BODY MASS INDEX IS " + Math.round(index) + ".");
40         System.out.println();
41     }
42 }
```

IOException

The throw IOException is used with the input and out put statement to handle errors when an invalid value has been entered by the user. For now just include lines 10 and the throws IOException on line 14. Later lessons will get into error handling.

```
10 import java.io.*;
11
12 public class BodyMass
13 {
14     public static void main(String[] args) throws IOException //Used with input and output
15     {
```

Declaring Variables

Declaring a variable declares the data type and names of the variables. In other words you are creating a storage place for a value to be used in the code. Variables can also be declared with a value as in the example below as in this example, `int count = 0`.

```
16          // declare and construct variables
17          String height, weight;
18          int inches, pounds;
19          double kilograms, meters, index;           Variables are declared in
```

lines 17, 18 and 19 in the application above. The table below list the types of variables used in Java.

Declare	Name	Description	Min. Value	Max. Value
boolean	Boolean	Holds two values: "true" and "false." Boolean values are <i>not</i> 1 or 0 as in C++. They must be set to "true" or "false", and they are not interchangeable with integers.	n/a	n/a
byte	Byte	The byte type holds one byte of data and is always an integer	-128	127
short	Short Integer	The short integer type holds 16-bits of data and is always an integer.	-32,768	32,767
int	Integer	The integer type hold 32-bits of data and is always an integer (Surprise, Surprise!).	-2,147,483,648	2,147,483,647
long	Long Integer	The long integer type holds 64-bits of data and is always an integer.	-1.845*10 ¹⁹	1.845*10 ¹⁹
float	Single-Precision	The single-precision data type is a floating point variable that occupies 32-bits of space.	-3.4*10 ³⁸	3.4*10 ³⁸
double	Double-Precision	The double-precision data type is a very large floating point variable that occupies 64-bits of space.	-1.7*10 ³⁰⁸	1.7*10 ³⁰⁸
char	Character	The character type holds one character (letter).	n/a	n/a

User Input – Streams and System Class

In this section the user is asked to input data. First on line 23 a title is sent to the screen then a blank line is sent for cosmetic reasons. On line 25 a prompt is sent. Note this is a `print` and not a `println` giving a prompt not just a line of text.

```
22          // print prompts and get input
23          System.out.println("\tTHE SUN FITNESS CENTER BODY MASS INDEX CALCULATOR");
24          System.out.println();
25          System.out.print("\t\tEnter your height to the nearest inch: ");
26          height = dataIn.readLine();
27          inches = Integer.parseInt(height);
28          System.out.print("\t\tEnter your weight to the nearest pound: ");
29          weight = dataIn.readLine();
30          pounds = Integer.parseInt(weight);
31
```

Lines 26 and 27 reads in the line as a string and then converts it to and the integer (inches) on line 27. The string must be converted to an integer so that math calculation can be preformed on inches. Remember the variables that were declared on lines 17 and 18.

```
16          // declare and construct variables
17          String height, weight;
18          int inches, pounds;
19          double kilograms, meters, index;
```

Lines 28 and 29 reads in the line as a string and then converts it to and the integer (pounds) on line 30. The string must be converted to an integer so that math calculation can be preformed on pounds.

Math Operators



- + Addition
- Subtraction
- * Multiplication
- / Division
- % Modulus (also called remainder)

Notice: no exponent! While there is no exponentiation symbol in Java, there is a `Math.pow()` method which will be discussed later.

Line 33 take the value stored in the variable (inches) and divides it by 39.36 to convert it to meters and line 34 takes the value stored in the variable (pounds) and divides it by 2.2 to convert it to kilograms. Why are we doing this? Just for fun!

```
32 // calculations
33 meters = inches / 39.36;
34 kilograms = pounds / 2.2;
35 index = kilograms / Math.pow(meters,2);
36
```

Math.pow() comes from the math class. **pow(double a, double b)**

- returns the first argument raised to the power of the second argument

`Math.pow(2,2):` 4.0

In this example if a user put in 60 inches and then it was divided by 39.36 it would equal 1.524390243902439024390243902439 and would be stored in the variable meters. Then on line 35 in the `Math.pow` method will be multiplied by the exponent of 2. In this example it would equal 2.323765615704937537180249851279 and now will be divided by the value stored in kilograms. In this example if a user put in 140 pounds and it is divided by 2.2 that you be 63.636363636363636363636363636364 kilograms. Now kilograms 63.636363636363636363636363636364 are divided by 2.323765615704937537180249851279 and equals 27.385018181818181818181818181818.

OutPut and Math.round Method

On line 39 the text "YOUR BODY MASS INDEX IS " is printed and the index is rounded and printed, plus the "." is printed. The "+" sign joins the text and the method together in a process called Concatenation.

```
37         // output
38         System.out.println();
39         System.out.println("\tYOUR BODY MASS INDEX IS " + Math.round(index) + ".");
40         System.out.println();
```

Using TextFields in an Applet

Review the code below. As you can see an applet takes considerably more code to accomplish the same task as the Body Index application above.

```
1  /*
2   Chapter 3: The Body Mass Index Calculator
3   Programmer: Wendy Plourde
4   Date:      October 20, 2007
5   Filename:  BodyMassApplet.java
6   Purpose:   This project calculates the body mass index based
7             on a person's height and weight.
8  */
9
10 import java.applet.*;
11 import java.awt.*;
12 import java.awt.event.*;
13
14 public class BodyMassApplet extends Applet implements ActionListener
15 {
16     //declare variables
17     Image logo; //declare an Image object
18     int inches, pounds;
19     double meters, kilograms, index;
20
21     //construct components
22     Label companyLabel = new Label("THE SUN FITNESS CENTER BODY MASS INDEX CALCULATOR");
23     Label heightLabel = new Label("Enter your height to the nearest inch: ");
24     TextField heightField = new TextField(15);
25     Label weightLabel = new Label("Enter your weight to the nearest pound: ");
26     TextField weightField = new TextField(10);
27     Button calcButton = new Button("Calculate");
28     Label outputLabel = new Label("Click the Calculate button to see your body mass index.");
29
30
31     //Set up for the applet
32     public void init()
33     {
34         setForeground(Color.red);
35         add(companyLabel);
36         add(heightLabel);
37         add(heightField);
38         add(weightLabel);
39         add(weightField);
40         add(calcButton);
41         calcButton.addActionListener(this); //register the addActionListener
42         add(outputLabel);
43         logo = getImage(getDocumentBase(), "logo.gif");
44     }
45
46     public void actionPerformed(ActionEvent e)
47     {
48
49         inches = Integer.parseInt(heightField.getText());
50         pounds = Integer.parseInt(weightField.getText());
51         meters = inches / 39.36;
52         kilograms = pounds / 2.2;
53         index = kilograms / Math.pow(meters,2);
54         outputLabel.setText("YOUR BODY MASS INDEX IS " + Math.round(index) + ".");
55     }
56
57     public void paint(Graphics g)
58     {
59         g.drawImage(logo,125,160,this);
60     }
61 }
```

When creating the applet so it can be displayed in a web page there are four kinds of objects you will create: an image, Labels, TextFields and Buttons. First you must import the correct packages to support the applet related methods. In this example line 10 imports java.applet.

```
10 import java.applet.*;
11 import java.awt.*;
12 import java.awt.event.*;
```

Next we have to implement the ActionListener to handle the events. Because the Body Index applet is interactive the program needs a way to handle presses on the keyboard and mouse clicks. On line 12 we must import java.awt.event to be able to use the event classes. Below is a list of class we will be using in the Body Mass Index applet.

Class	Example of Event	Associated Listener Interface
ActionEvent	User click a button, presses the ENTER key, or chooses a menu item	ActionListener
ItemEvent	User selects or deselects an item such as a checkbox or and option button	ItemListener
KeyEvent	User presses a key	KeyListener
MouseEvent	User performs a mouse action, such as select, drag or enter	MouseListener
TextEvent	User changes text in text box	TextListener
WindowEvent	Window changes status by opening, closing, or performing some other action	WindowListener

On line 14 we first have to extend applet to make this usable on the web and then implement the ActionListener. The key word **implements** is used in this class header to specify which listener interface will be used.

```
14 public class BodyMassApplet extends Applet implements ActionListener
15 {
16     //declare variables
17     Image logo; //declare an Image object
18     int inches, pounds;
19     double meters, kilograms, index;
20 }
```

Now we can declare the variables. When the variables are declared after the header class, as in this case, this means the entire program can use the variables and they are not restricted to just one method.

In this next section is where we construct all of the components of the applet. This is kind of like creating variables.

```
21 //construct components
22 Label companyLabel = new Label("THE SUN FITNESS CENTER BODY MASS INDEX CALCULATOR");
23 Label heightLabel = new Label("Enter your height to the nearest inch: ");
24     TextField heightField = new TextField(15);
25 Label weightLabel = new Label ("Enter your weight to the nearest pound: ");
26     TextField weightField = new TextField(10);
27 Button calcButton = new Button("Calculate");
28 Label outputLabel = new Label("Click the Calculate button to see your body mass index.");
29
```

In this next section the `init()` method is used to add all of the components to the applet interface. When it is defined as **public void** that means it does not return any values.

```
31     //Set up for the applet
32     public void init()
33     {
34         setForeground(Color.red);
35         add(companyLabel);
36         add(heightLabel);
37         add(heightField);
38         add(weightLabel);
39         add(weightField);
40         add(calcButton);
41         calcButton.addActionListener(this); //register the addActionListener
42         add(outputLabel);
43         logo = getImage(getDocumentBase(), "logo.gif");
44     }
45
```

The `setForeground()` method used in this section is to make the applet visually appealing, such as, the text color. The `add()` method is used to place each field, label or button on the applet. The `calcButton` is registered as the `ActionListener` so when the button is clicked the program will know how to respond. On line 43 `getImage()` calls `getDocumentBase` which allows the applet to pull the image from the current folder in which the applet is stored.

When a user clicks on the Calculate Button this final piece of code the values are read in for the Textfields and they are parsed or converted to integers and the answer is displayed in the `outputLabel`.

```
46     public void actionPerformed(ActionEvent e)
47     {
48
49         inches = Integer.parseInt(heightField.getText());
50         pounds = Integer.parseInt(weightField.getText());
51         meters = inches / 39.36;
52         kilograms = pounds / 2.2;
53         index = kilograms / Math.pow(meters,2);
54         outputLabel.setText("YOUR BODY MASS INDEX IS " + Math.round(index) + ".");
55     }
```

The final method `paint()` is used to draw the image on the screen. Remember you will need to find an image and place it in the folder where you applet is.

Creating the HTML

This file is used to run the `BodyMassApplet.class` and can be viewed in the browser. You could now run this file on a web site. All is needed is the HTML and the compiled java code, which is the `.class` file. No mystery here! The only thing to remember is to make sure you call the right class and define the height and width of the window will be.

```
1 <HTML>
2 <APPLET CODE = "BodyMassApplet.class" WIDTH = 375 HEIGHT = 300>
3 </APPLET>
4 </HTML>
```

Programming Projects

1. Create the Body Mass Index application.
2. Create the Body Mass Index applet and create the HTML file to call the class file.