

Chapter 2

Variables and Operators

2.1 The Assignment Operator

In Java, the symbol `=` is called the *assignment operator*, and it is **not** the same as the equal sign in algebra. Here's how the assignment operator works:

```
int n;  
n = 7 * 8 + 9;
```

1. Look at the right-hand side of the assignment operator first: `7 * 8 + 9`
2. Evaluate the right-hand side completely. The right hand side works out to 65.
3. Put that result into the variable on the left hand side of the assignment operator.

Whenever you see the assignment operator, don't read it with the "e-word." Instead, when you see a statement like this: `years = 23;`, read it as one of the following:

- `years` becomes 23
- `years` gets the value 23
- `years` is assigned 23

- `years` is set to 23

To bring home the fact that Java's assignment operator `=` doesn't work like the same symbol in algebra, consider this code:

```
int years = 23;
years = years + 1;
```

The first assignment takes the 23 and assigns it to the integer variable `years`.

The second assignment statement is absurd in algebra, but makes perfect sense in Java, following the rules for the assignment operator:

- Look at the right hand side and work it out completely. Right now, `years` contains 23, and $23 + 1$ works out to 24.
- Now take that result and put it into the variable on the left hand side. This replaces the value of variable `years` with the new value of 24.

You can read these statements as: “integer `years` becomes 23” and “`years` is assigned $\text{years} + 1$.”

2.2 Exercises

Exercise 2.1 Use Java to find your approximate age in days.

1. Create a new program name *AgeInDays.java*. Copy or type in something like the Hello World program and make sure you can compile and run it.
2. Write a program that creates variables named `years` and `days` which represent your age in years and your age in days. Both of these are integers.
3. Set the `years` to your current age in years.
4. Calculate the `days` as `years` times 365.
5. Print the age in years and days, properly labeled. Here is what the output might look like:

```
I am 25 years old.
That is about 9125 days.
```

Exercise 2.2 The point of this exercise is to use Java’s arithmetic operators to calculate the *dew point*—the temperature at which water begins to condense out of the air. The dew point formula is more complicated than the one in the preceding exercise.

1. Write a program that creates variables named `temperature` and `relHumidity` which represent the temperature in degrees Celsius and the relative humidity as a percentage from 0 to 100. These will be `double` values. Assign values to those variables that represent a temperature of 17°C and a relative humidity of 30%.
2. Display the value of each variable on a line by itself. This is an intermediate step that is useful for checking that everything is working so far. Compile and run your program before moving on.
3. Calculate the dew point temperature using this approximate formula:

$$dewPoint = temperature - \frac{100 - relHumidity}{5}$$

and display the value of the result. The `dewPoint` variable will also be a `double`.

4. Display the result of the calculation, properly labeled. Here is what the output might look like. Your output does not have to match this exactly, but it must reflect the same information.

```
For air temperature of 17.0 degrees Celsius  
and relative humidity of 25.0%  
The dew point is 2.0 degrees Celsius.
```

Exercise 2.3 This exercise uses a fairly large formula to calculate the *heat index*, a measurement of how hot it feels when the relative humidity is factored in with the temperature. The purpose of this exercise is to get you comfortable with doing complex calculations in Java.

1. Create a new program named *HeatIndex.java*.
2. Create variables named `t` and `rh`, where `t` stands for the temperature and `rh` stands for the relative humidity. Set the temperature to 30 degrees Celsius and the relative humidity to 75%.

3. Ordinarily we would use longer variable names like `temperature` and `relHumidity` as in Exercise 2.2, but the length of the formula makes it easier to type when using short names. Write a comment in your Java program that says that you are using `t` to stand for the temperature in degrees Celsius and `rh` for relative humidity.
4. Calculate the heat index using this formula:

$$\begin{aligned} \text{heatIndex} = & -8.78469475556 + 1.61139411 \cdot t + 2.338548839 \cdot rh - \\ & 0.14611605 \cdot t \cdot rh - 0.012308094 \cdot t \cdot t - \\ & 0.0164248278 \cdot rh \cdot rh + 0.002211732 \cdot t \cdot t \cdot rh + \\ & 0.00072546 \cdot t \cdot rh \cdot rh - 0.000003582 \cdot t \cdot t \cdot rh \cdot rh \end{aligned}$$

Don't put this entire equation on one line in Java. Instead, split it into several lines as shown here, or use temporary variables to hold the parts of the formula.

5. Display the result of the calculation. Here is what the output might look like.

```
For air temperature of 30.0 degrees Celsius  
and relative humidity of 75.0%  
The heat index is 36.299647994439965 degrees Celsius.
```