JAVA Textbook

Chapter 2

Variables, Constants, Operators, and Writing Programs Using Sequential Statements

Objectives

In this chapter, you will learn about:

- Variables in Java
- Constants in Java
- Operators and expressions in Java
- Sequential statements, comments, and interactive input statements in Java

Variables in Java

- A variable is a named location in the computer's memory whose content can vary.
  - Sometimes called an identifier
- A variable in a program is used to store values that often change.
- In Java, you must declare variables before using them.
  - Name the variable
  - Specify its data type

Variable Names

- Can consist of letters, numerical digits, a dollar sign, and the underscore character
- Cannot begin with a digit
- Cannot use a Java keyword for a variable name
- Cannot include spaces
- Are case sensitive
- Meaningful names are preferred

Java Data Type

- Data type determines the amount of memory allocated for a variable, and the type of data that can be stored in the variable.
- Primitive data types refer to the most basic types.
  - Numeric: short, int, long, float, double
  - Others: byte, char, and boolean
- Data types used often in this course
  - int and double for numeric
  - Boolean for logical
  - String (which is actually an object, not a data

Declaring and Initializing Variables

- In Java, must declare variables before using them.
- Java syntax for a variable declaration:

  dataType variableName;
- You can also initialize a Java variable when you declare it.

  dataType variableName = initialValue;
- Numeric variables automatically initialized to zero (0) unless you specify a different value.

Example:
Variable Declaration, Initialization

```
int counter;
int counter = 8;
int counter, value;
double salary;
double cost = 12.95;
String firstName;
String homeAddress = "123 Main Street";
```

Constants in Java

- A constant is used to store a value that never change.
  - Names: unnamed constants, named constants
  - Data types: numeric constant, string constant
- Unnamed constants
  - Use numeric value (e.g. 18) or string (e.g., “Mercer University”) directly without a name.
- Named constants
  - Similar to variable, but only assigned value once.
  - Syntax: *final dataType constantName = constantValue;*

Arithmetic Operators

Operator Name and Symbol	Example	Comment
Addition +	`num1 + num2`	
Subtraction -	`num1 - num2`	
Multiplication *	`num1 * num2`	
Division /	`15/2`	Integer division; result is 7; fraction is truncated
	`15.0 / 2.0`	Floating-point division; result is 7.5
	`15.0 / 2`	Floating-point division because one of the operands is a floating-point number; result is 7.5
Modulus %	`hours % 24`	Performs division and finds the remainder; result is 1 if the value of `hours` is 25
Unary plus +	`+num1`	Maintains the value of the expression; if the value of `num1` is 3, then `+num1` is 3
Unary minus -	`-(num1 - num2)`	If value of `(num1 - num2)` is 10, then `-(num1 - num2)` is –10

Table 2-2 Java arithmetic operators

- Used to perform arithmetic calculations.

Arithmetic Expressions

If:

```
int num1 = 3, num2 = 20;
```

Then:

Expression	Value	Explanation
`num1 + num2`	23	3 + 20 = 23
`num1 - num2`	−17	3 − 20 = −17
`num2 % num1`	2	20 / 3 = 6 remainder 2
`num1 * num2`	60	3 * 20 = 60
`num2 / num1`	6	20 / 3 = 6 (remainder is truncated)
`-num1`	−3	Value of `num1` is 3, therefore `-num1` is −3

Table 2-3 Expressions and values

- You can combine arithmetic operators and variables to create expressions.
  - Each expression will give a result, which is a value

Assignment Operators

Operator Name and Symbol	Example	Comment
Assignment =	`count = 5;`	Places the value on the right side into the memory location named on the left side.
Initialization =	`int count = 5;`	Places the value on the right side into the memory location named on the left side when the variable is declared.
Assignment +=	`num += 20;`	Equivalent to `num = num + 20;`
Assignment -=	`num -= 20;`	Equivalent to `num = num - 20;`
Assignment *=	`num *= 20;`	Equivalent to `num = num * 20;`
Assignment /=	`num /= 20;`	Equivalent to `num = num / 20;`
Assignment %=	`num %= 20;`	Equivalent to `num = num % 20;`

Table 2-4 Java assignment operators

- Used to assign a value to a variable.

Precedence and Associativity

Operator Name	Operator Symbol	Order of Precedence	Associativity
Parentheses	()	First	Left to right
Unary	- +	Second	Right to left
Multiplication, division, and modulus	* / %	Third	Left to right
Addition and subtraction	+ -	Fourth	Left to right
Assignment	= += -=	Fifth	Right to left
	*= /= %=		

Table 2-5 Order of precedence and associativity

- Precedence: Order of operations to be performed
- Associativity: Order of operations of the same precedence to be performed

Sequential Statements & Comments

- Sequential statements (or sequence): a series of statements that must be performed in sequential order, one after another.
- Comments: serve as documentation, explaining the code to whoever might read it.
  - Not executed.
  - Well-written, meaningful comments are always expected.
  - Two commenting styles in Java.
    - Type two forward slash characters // at the beginning of each line of comments: useful to mark a single line as a comment.
    - Enclose a block of lines with the characters /* and */: useful to mark multiple lines as a comment.

Interactive Input Statements

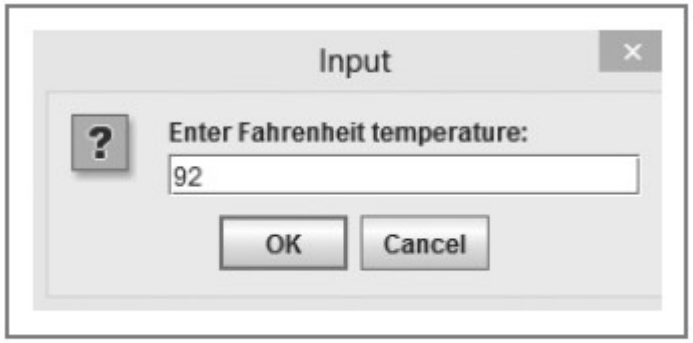

Figure 2-2 An input dialog box

```
/* Temperature.java - This program converts a Fahrenheit
   temperature to Celsius.
   Input: Interactive
   Output: Fahrenheit temperature followed by Celsius
   temperature
*/
import javax.swing.JOptionPane; // Import JOptionPane class
public class Temperature
{
   public static void main(String args[])
   {
      String fahrenheitString;
      double fahrenheit;
      double celsius;
      // Get interactive user input
      fahrenheitString = JOptionPane.showInputDialog(
                         "Enter Fahrenheit temperature: ");
      // Convert String to double
      fahrenheit = Double.parseDouble(fahrenheitString);
      // Calculate Celsius equivalent
      celsius = (fahrenheit - 32.0) * (5.0 / 9.0);
      // Output
      System.out.println("Fahrenheit temperature:" +
                         fahrenheit);
      System.out.println("Celsius temperature:" + celsius);
      // End program
      System.exit(0);
   }
}
```

- Interactive input statements: statements that ask, or prompt, the user to input data

Summary

- Variables need to be declared before being used.
- Both unnamed and named constants can be used, where a named constant also needs to be declared before being used.
- Operators can be used to form expressions for calculations, etc.
- Sequential statements, comments, and interactive input statements are most commonly used in a typical Java program

Thank You!