1. **Mod (Modulo) %**
   performs remainder(modulus) division with integers
   10 \% 3 -> 1
   3 \% 10 -> 3 (Note that when a smaller number is divided by a larger, the remainder is always equal to the smaller number because the larger number goes into the smaller zero times:

   \[
   \begin{array}{c}
   0 \text{ R } 3 \\
   10 \text{ ) 3}
   \end{array}
   \]
   3 is the remainder

   what is 5 \% 3? 53 \% 10? 17 \% 18? 17 \% 2? 18 \% 2?

2. **Combined assignment operators**

   When you want the same variable name on both sides of the assignment operator,

   \[
   \text{nTotalHours} = \text{nTotalHours} + \text{nHours};
   \]

   this is a shortcut syntax:

   \[
   \text{nTotalHours} += \text{nHours};
   \]

3. **Declaring constants using the reserved word final; capitalize the identifier with underscores to separate words.**

   Constants cannot change their value, unlike variables.
Final double MAX_TEMP = 98.6;

Trying to assign a new value to would produce a compiler error. Advantage: the value is set once at the beginning of the program but can be used in many places. One revision to the original declaration changes the value in each place.

4. Scope of Variables
   The scope means the part of the program where the variable name has meaning, so that when you refer to that name, you are accessing its location in memory. Any variable declared inside a method is called a local variable. Its scope spans from its declaration, to the closing brace of the method. We cannot have two variables with the same name and scope. Java is case sensitive, so num1 and Num1 are 2 different variables.


6. Input and Output of numbers and strings

Scanner class new to Java 1.5

First import the Scanner class into your program:

import java.util.Scanner; // util is a package of basic classes

We use standard syntax to instantiate an object of the Scanner class:

Scanner input = new Scanner(System.in); // associates the Scanner object with the System.in predefined object (remember System.out)

The author used the variable name keyboard, while I’m using input, just to show it is an identifier, not a keyword. Here input is called a variable reference because it holds the address of an object in memory, not a single value like a variable. (draw picture). We can now call any method of the Scanner class on input. It is our connection to all the members of the Scanner class (properties and methods)

7. Methods of the Scanner class
for numbers:
   nextByte
   nextDouble
   nextFloat
   nextInt
   nextLong
   nextShort
for Strings:
   nextLine – to get a String input

Let’s trace the example program on p. 94, Payroll.java, to see how the Scanner class methods can be used to get input from the user.

Every key the user presses is stored in the Keyboard buffer. Let’s look at the program on p.98, CorrectedInputProblem, to examine how to clear the keyboard buffer of an unwanted character.

8. Methods of the JOptionPane class
JOptionPane creates Dialog boxes for simple output, or for both input and output. When we create dialog boxes in an application, we are writing a GUI application. (see earlier handout on program types).

Note: GUI applications cannot run on hills, though they can be compiled there.

For output only, use the showMessageDialog method. No value is returned.

First import the JOptionPane class into your program:
import javax.swing.JOptionPane; // swing is a package of graphics-related classes

We use non-standard syntax in that we do not use the new operator to instantiate a JOptionPane object before we call its method.

JOptionPane.showMessageDialog(null, “Hello”); // don’t worry about null now
Another kind of dialog box from the JOptionPane class, for input, is JOptionPane.showInputDialog - it prompts the user for input and returns the user's input as a string, which we store in a variable. Notice we assign the string returned by the method.

ex. String firstNumber = JOptionPane.showInputDialog("Enter first integer");

Converting String input to Numbers

   Integer.parseInt() converts a string to an int only if the string contains digits. (otherwise it throws an exception, for example “15a”
   It returns the integer equivalent, which we store in a variable.
   ex. int number1 = Integer.parseInt(firstNumber);

If time, let’s look at PayrollDialog.java on page 104. Not the use of the + operator for String concatenation.

Section 2.15 is important, Common Errors. Also look at linked page on Resources page.

Sample Question for test 1:

What is the value of result after the following Java statements execute?
int a = 6, b = 3, c = 2, d = 43;
result = d % b * c + b % a + c;

what is the order of operations?