Karrlem Lewis

Prof Yan

CSC 110 ch.2 Notes

February 13th, 2016

* Java source code files end with .java.
* **public** is a Java keyword, and it must be written in all lowercase letters. It is known as an access specifier, and it controls where the class may be accessed from. The public specifier means access to the class is unrestricted.
* **class**, which must also be written in lowercase letters, is a Java keyword that indicates the beginning of a class definition.

public static void main(String[] args)

* This line is known as a method header. It marks the beginning of a method. A method can be thought of as a group of one or more programming statements that collectively has a name.

Special characters

// **Double slash**  Marks the beginning of a comment

( ) **Opening and closing parentheses** Used in a method header

{ } **Opening and closing braces** Encloses a group of statements, such as the contents of a class or a method

" " **Quotation marks**  Encloses a string of characters, such as a message that is to be printed on the screen

; **Semicolon Marks** the end of a complete programming statement

The **print** and **println** methods are used to display text output. They

are part of the Java API

System.out.println(“”);

Common escape sequences

\n **Newline**  Advances the cursor to the next line for subsequent printing

\t **Horizontal tab** Causes the cursor to skip over to the next tab stop

\b **Backspace**  Causes the cursor to back up, or move left, one position

\r **Return**  Causes the cursor to go to the beginning of the current line, not the next line

\\ **Backslash**  Causes a backslash to be printed

\' **Single quote**  Causes a single quotation mark to be printed

\" **Double quote** Causes a double quotation mark to be printed

A variable is a named storage location in the computer’s memory. A literal

is a value that is written into the code of a program.

int value;

This is called a variable declaration. Variables must be declared before they can be used. A variable declaration tells the compiler the variable’s name and the type of data it will hold.

This line indicates the variable’s name is value. The word int stands for integer, so value

will only be used to hold integer numbers. Notice that variable declarations end with a

semicolon.

value = 5;

This is called an assignment statement. The equal sign is an operator that stores the value

on its right (in this case 5) into the variable named on its left. After this line executes, the

value variable will contain the value 5.

When the + operator is used with strings, it is known as the string concatenation operator.

To concatenate means to append, so the string concatenation operator appends one string

to another.

System.out.println("This is " + "one string.");

This statement will print: This is one string.

The + operator produces a string that is the combination of the two strings used as its operands. You can also use the + operator to concatenate the contents of a variable to a string.

number = 5;

System.out.println("The value is " + number);

Literal Type of Literal

20 Integer literal

“Today we sold ” String literal

“ bushels of apples.” String literal

An identifier is a programmer-defined name that represents some element of a program. Variable names and class names are examples of identifiers. You may choose your own variable names and class names in Java, as long as you do not use any of the Java keywords.

Variable Name Legal or Illegal?

dayOfWeek Legal

3dGraph Illegal because identifiers cannot begin with a digit

june1997 Legal

mixture#3 Illegal because identifiers may use only alphabetic letters, digits,

underscores, or dollar signs

week day Illegal because identifiers cannot contain spaces

Data Type Size Range

byte 1 byte Integers in the range of −128 to +127

short 2 bytes Integers in the range of −32,768 to +32,767

int 4 bytes Integers in the range of −2,147,483,648 to +2,147,483,647

long 8 bytes Integers in the range of −9,223,372,036,854,775,808 to

+9,223,372,036,854,775,807

float 4 bytes Floating-point numbers in the range of ±3.4 × 10−38 to ±3.4 × 1038, with 7 digits of accuracy

double 8 bytes Floating-point numbers in the range of ±1.7 × 10−308 to ±1.7 × 10308, with 15 digits of accuracy

DataType is the name of the data type and VariableName is the name of the variable.

byte inches;

int speed;

short month;

float salesCommission;

double distance;

number = 1,257,649; // ERROR!

This statement must be written as:

number = 1257649; // Correct.

Decimal Notation Scientific Notation E Notation

247.91 2.4791 × 10^2 2.4791E2

0.00072 7.2 × 10^−4 7.2E–4

2,900,000 2.9 × 106 2.9E6

The boolean data type allows you to create variables that may hold one of two possible

values: true or false.

Operator Meaning Type Example

+ Addition Binary total = cost + tax;

− Subtraction Binary cost = total − tax;

\* Multiplication Binary tax = cost \* rate;

/ Division Binary salePrice = original / 2;

% Modulus Binary remainder = value % 3;

Highest Precedence → - (unary negation) Right to left

\* / % Left to right

Lowest Precedence → + − Left to right

Ex:

Expression Value

5 + 2 \* 4 13

10 / 2 - 3 2

8 + 12 \* 2 - 4 28

4 + 17 % 2 - 1 4

6 - 3 \* 2 + 7 - 1 6

Expression Value

(5 + 2) \* 4 28

10 / (5 − 3) 5

8 + 12 \* (6 − 2) 56

(4 + 17) % 2 −1 0

(6 − 3) \* (2 + 7) / 3 9

/\*

2 PROGRAM: Comment2.java

3 Written by Herbert Dorfmann

4 This program calculates company payroll

5

\*/

/\*

\* This program demonstrates the

\* way to write comments.

\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// This program demonstrates the \*

// way to write comments. \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

////////////////////////////////////

// This program demonstrates the

// way to write comments.

////////////////////////////////////

//-----------------------------------

// This program demonstrates the

// way to write comments.

//-----------------------------------

The showMessageDialog method is used to display a message dialog.

JOptionPane.showMessageDialog(null, "Hello World");

import javax.swing.JOptionPane;

2

3 /\*\*

4 This program demonstrates using dialogs with

5 JOptionPane.

6 \*/

7

8 public class NamesDialog

9 {

10 public static void main(String[] args)

11 {

12 String firstName; // The user's first name

13 String middleName; // The user's middle name

14 String lastName; // The user's last name

15

16 // Get the user's first name.

17 firstName =

18 JOptionPane.showInputDialog("What is " +

19 "your first name? ");

20

21 // Get the user's middle name.

22 middleName =

23 JOptionPane.showInputDialog("What is " +

24 "your middle name? ");

25

26 // Get the user's last name.

27 lastName =

28 JOptionPane.showInputDialog("What is " +

29 "your last name? ");

30

31 // Display a greeting

32 JOptionPane.showMessageDialog(null, "Hello " +

33 firstName + " " + middleName +

34 " " + lastName);

35 System.exit(0);

36 }

37 }

Common Errors to avoid

• Mismatched braces, quotation marks, or parentheses. In this chapter you saw that the

statements making up a class definition are enclosed in a set of braces. Also, you saw

that the statements in a method are also enclosed in a set of braces. For every opening

brace, there must be a closing brace in the proper location. The same is true of doublequotation

marks that enclose string literals and single-quotation marks that enclose

character literals. Also, in a statement that uses parentheses, such as a mathematical

expression, you must have a closing parenthesis for every opening parenthesis.

• Misspelling key words. Java will not recognize a key word that has been misspelled.

• Using capital letters in key words. Remember that Java is a case-sensitive language,

and all key words are written in lowercase. Using an uppercase letter in a key word is

the same as misspelling the key word.

• Using a key word as a variable name. The key words are reserved for special uses; they

cannot be used for any other purpose.

• Using inconsistent spelling of variable names. Each time you use a variable name, it

must be spelled exactly as it appears in its declaration statement.

• Using inconsistent case of letters in variable names. Because Java is a case-sensitive language,

it distinguishes between uppercase and lowercase letters. Java will not recognize

a

variable name that is not written exactly as it appears in its declaration statement.

• Inserting a space in a variable name. Spaces are not allowed in variable names. Instead

of using a two-word name such as gross pay, use one word, such as grossPay.

• Forgetting the semicolon at the end of a statement. A semicolon appears at the end of

each complete statement in Java.

• Assigning a double literal to a float variable. Java is a strongly typed language,

which means that it only allows you to store values of compatible data types in variables.

All floating-point literals are treated as doubles, and a double value is not compatible

with a float variable. A floating-point literal must end with the letter f or F in

order to be stored in a float variable.

• Using commas or other currency symbols in numeric literals. Numeric literals cannot

contain commas or currency symbols, such as the dollar sign.

• Unintentionally performing integer division. When both operands of a division statement

are integers, the statement will result in an integer. If there is a remainder, it will

be discarded.

• Forgetting to group parts of a mathematical expression. If you use more than one

operator in a mathematical expression, the expression will be evaluated according to

the order of operations. If you wish to change the order in which the operators are

used, you must use parentheses to group part of the expression.

• Inserting a space in a combined assignment operator. A space cannot appear between

the two operators that make a combined assignment operator.

• Using a variable to receive the result of a calculation when the variable’s data type is

incompatible with the data type of the result. A variable that receives the result of a

calculation must be of a data type that is compatible with the data type of the result.

• Incorrectly terminating a multi-line comment or a documentation comment. Multiline

comments and documentation comments are terminated by the \*/ characters.

Forgetting to place these characters at a comment’s desired ending point, or accidentally

switching the \* and the /, will cause the comment not to have an ending point.

• Forgetting to use the correct import statement in a program that uses the Scanner

class or the JOptionPane class. In order for the Scanner class to be available to your

program, you must have the import java.util.Scanner; statement near the top of

your program file. In order for the JOptionPane class to be available to your program,

you must have the import javax.swing.JOptionPane; statement near the top of the

program file.

• When using an input dialog to read numeric input, not converting the showInput-

Dialog method’s return value to a number. The showInputDialog method always

returns the user’s input as a string. If the user enters a numeric value, it must be converted

to a number before it can be used in a math statement.