Chapter 2

How to start writing Java code
Objectives

Applied

• Use NetBeans to create a project for a Java application and use NetBeans to work with the source code for that project.
• Write the code for a class that contains a main method.
• Use single-line comments and block comments to document code.
• Write code that uses the methods of the System.out object to print data to the console.
• Write code that works with variables of the int and double types.
• Write code that uses the arithmetic operators to add, subtract, multiply, and divide numbers.
• Write code that works with variables of the String type.
• Use escape sequences to include special characters in a string.
Objectives (cont.)

Knowledge

• Given a list of names, identify the ones that follow the naming recommendations for classes presented in this chapter.
• Given a list of names, identify the ones that follow the naming recommendations for variables presented in this chapter.
• Distinguish between the int and double data types.
A class that has a main method

```java
package murach.test;

public class CodeTesterApp {
    public static void main(String[] args) {
        // Code here
    }
}
```

1. Package statement
2. Class declaration
3. Main method declaration
4. End of main method
5. End of class
The folder and filename for this class
murach/test/CodeTesterApp.java

The guidelines for naming a class

- Start the name with a capital letter.
- Start every word within a class name with an initial cap.
- Use letters and digits only.
A class that includes statements and comments

/*
 * Author:  J. Murach
 * Purpose: This application displays some text on the console.
 *          It can be used as a starting point for testing code.
 */
package murach.test;          // package statement

public class CodeTesterApp {    // start CodeTesterApp class

    public static void main(String args[]) {
        // start main method
        // display a welcome message
        System.out.println("Welcome to the Code Tester");
        System.out.println();     // print a blank line
        System.out.println("Bye!");
    }                           // end main method
}                                 // end CodeTesterApp class
Two methods of the System.out object

`println(data)`
`print(data)`
A class that prints output to the console

```java
package murach.test;

public class CodeTesterApp {

    public static void main(String args[]) {
        // display a welcome message
        System.out.println("Welcome to the Code Tester");
        System.out.println();

        // display a goodbye message
        System.out.println("Bye!");
    }
}
```

The console

```
Welcome to the Code Tester

Bye!
```
Two of the eight primitive data types

int
double
How to declare and initialize a variable in two statements

Syntax

type variableName;
variableName = value;

Example

double productPrice; // declaration statement
productPrice = 14.95; // assignment statement
How to declare and initialize a variable in one statement

Syntax

type variableName = value;

Example

double productPrice = 14.95 // initialize double variable
Naming recommendations for variables

• Start variable names with a lowercase letter and capitalize the first letter in all words after the first word.
• Each variable name should be a noun or a noun preceded by one or more adjectives.
• Try to use meaningful names that are easy to remember.
# Java keywords

<table>
<thead>
<tr>
<th>boolean</th>
<th>if</th>
<th>interface</th>
<th>class</th>
<th>true</th>
</tr>
</thead>
<tbody>
<tr>
<td>char</td>
<td>else</td>
<td>package</td>
<td>volatile</td>
<td>false</td>
</tr>
<tr>
<td>byte</td>
<td>final</td>
<td>switch</td>
<td>while</td>
<td>throws</td>
</tr>
<tr>
<td>float</td>
<td>private</td>
<td>case</td>
<td>return</td>
<td>native</td>
</tr>
<tr>
<td>void</td>
<td>protected</td>
<td>break</td>
<td>throw</td>
<td>implements</td>
</tr>
<tr>
<td>short</td>
<td>public</td>
<td>default</td>
<td>try</td>
<td>import</td>
</tr>
<tr>
<td>double</td>
<td>static</td>
<td>for</td>
<td>catch</td>
<td>synchronized</td>
</tr>
<tr>
<td>int</td>
<td>new</td>
<td>continue</td>
<td>finally</td>
<td>const</td>
</tr>
<tr>
<td>long</td>
<td>this</td>
<td>do</td>
<td>transient</td>
<td>goto</td>
</tr>
<tr>
<td>abstract</td>
<td>super</td>
<td>extends</td>
<td>instanceof</td>
<td>null</td>
</tr>
</tbody>
</table>

## Naming rule

- Keywords are reserved for use by Java. You can’t use them as the name for a variable.
The syntax for an assignment statement

variableName = value;

Code that declares and initializes two variables

int quantity = 0;                   // quantity is 0
int maxQuantity = 100;

Code that assigns new values to a variable

quantity = 10;                     // quantity is now 10
quantity = maxQuantity;            // quantity is now 100

Code that tries to declare a variable a second time

int quantity = 99;                 // ERROR: quantity already declared

Code that uses incorrect case

maxquantity = 200;                  // ERROR: incorrect case
## The basic arithmetic operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Addition</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
</tr>
</tbody>
</table>
Code that calculates a total amount

double productPrice = 9.99;
int quantity = 2;
double total = productPrice * quantity;     // 19.98

Code that calculates sales tax

double subtotal = 250.00;
double taxPercent = .075;                    // 7.5%
double taxAmount = subtotal * taxPercent;   // 18.75
double grandTotal = subtotal + taxAmount;    // 268.75

Code that calculates an average

double sum = 206.75;
int count = 3;
double average = sum / count;              // 68.91666666666667
Code that divides two int values

```java
int miles = 9;
int hour = 2;
int milesPerHour = miles / hour;             // 4
```

Code that uses multiple operators

```java
double width = 4.25;
double length = 8.5;
double perimeter = 2 * width + 2 * length;   // 25.5
```
How to declare and initialize a String variable

Syntax

```java
String variableName = value;
```

Examples

```java
String message1 = "Invalid data entry.";
String message2 = "";                 // an empty string
String message3 = null;              // a null string
```
How to join strings

String firstName = "Bob";
String lastName = "Smith";
String name = firstName + " " + lastName;    // Bob Smith

How to join a string and a number

double price = 14.95;
String message = "Price: " + price;    // Price: 14.95
## Common escape sequences

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>\n</td>
<td>New line</td>
</tr>
<tr>
<td>\t</td>
<td>Tab</td>
</tr>
<tr>
<td>\r</td>
<td>Return</td>
</tr>
<tr>
<td>&quot;</td>
<td>Quotation mark</td>
</tr>
<tr>
<td>\</td>
<td>Backslash</td>
</tr>
</tbody>
</table>
The new line character

"Code:  JSP\nPrice: $49.50"

Displayed on the console

<table>
<thead>
<tr>
<th>Code</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>$49.50</td>
</tr>
</tbody>
</table>
The new line, return, and tab characters

"Code:\tJSP\r\nPrice:\t$49.50"

Displayed on the console

<table>
<thead>
<tr>
<th>Code:</th>
<th>JSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price:</td>
<td>$49.50</td>
</tr>
</tbody>
</table>
A syntax error due to quotation marks

"Type "x" to exit"  // syntax error - ')' expected

Quotation marks

"Type "x" to exit"

Displayed on the console

Type "x" to exit
A syntax error due to backslashes

"C:\java\files"  // illegal escape character

Backslashes

"C:\\java\\files"

Displayed on the console

C:\java\files
The Code Tester application

Welcome to the Code Tester

Code: java
Price: 49.5
Quantity: 2
Total: 99.0

Bye!
The CodeTesterApp class

package murach.test;

public class CodeTesterApp {

    public static void main(String args[]) {

        // display a welcome message
        System.out.println("Welcome to the Code Tester");
        System.out.println();

        // hard code three values
        String productCode = "java";
        double price = 49.50;
        int quantity = 2;
    }
The CodeTesterApp class (cont.)

    // perform a calculation
    double total = price * quantity;

    // display the output
    String message =
        "Code:    " + productCode + "\n" +
        "Price:   " + price + "\n" +
        "Quantity: " + quantity + "\n" +
        "Total:   " + total + "\n";
    System.out.println(message);

    // display a goodbye message
    System.out.println("Bye!");
}