Chapter 2

Introduction to Java programming

Keywords

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

A public class that contains a main method

```
public class InvoiceApp                 // declare the class
{                                       // begin the class
    public static void main(String[] args)
    {
        System.out.println(
            "Welcome to the Invoice Total Calculator");
    }
}                                       // end the class
```

The same class with different brace placement

```
public class InvoiceApp {  // declare and begin the class
   public static void main(String[] args){
      System.out.println(
          "Welcome to the Invoice Total Calculator");
   }
}                             // end the class
```

The rules for naming a class

- Start the name with a capital letter.
- Use letters and digits only.
- Follow the other rules for naming an identifier.

Recommendations for naming a class

- Start every word within a class name with an initial cap.
- Each class name should be a noun or a noun that’s preceded by one or more adjectives.

How to declare and initialize a variable in one statement

Syntax

```
type variableName = value;
```

Examples

```
int scoreCounter = 1;   // initialize an integer variable
double unitPrice = 14.95; // initialize a double variable
```

How to code assignment statements

```
int quantity = 0;                       // initialize an
                                        // integer variable
int maxQuantity = 100;                  // initialize another
                                        // integer variable

// two assignment statements
quantity = 10;                          // quantity is now 10
quantity = maxQuantity;                 // quantity is now 100
```

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.

The basic operators for arithmetic expressions

Operator	Name
`+`	Addition
`-`	Subtraction
`*`	Multiplication
`/`	Division

Statements that use simple arithmetic expressions

```
// integer arithmetic
int x = 14;
int y = 8;
int result1 = x + y;          // result1 = 22
int result2 = x - y;          // result2 = 6
int result3 = x * y;          // result3 = 112
int result4 = x / y;          // result4 = 1

// double arithmetic
double a = 8.5;
double b = 3.4;
double result5 = a + b;       // result5 = 11.9
double result6 = a - b;       // result6 = 5.1
double result7= a * b;        // result7 = 28.9
double result8 = a / b;       // result8 = 2.5
```


Statements that increment a counter variable

```
int invoiceCount = 0;
invoiceCount = invoiceCount + 1;      // invoiceCount = 1
invoiceCount = invoiceCount + 1;      // invoiceCount = 2
```

Statements that add amounts to a total

```
double invoiceAmount1 = 150.25;
double invoiceAmount2 = 100.75;
double invoiceTotal = 0.0;
invoiceTotal = invoiceTotal + invoiceAmount1;
                                      // invoiceTotal = 150.25
invoiceTotal = invoiceTotal + invoiceAmount2;
                                      // invoiceTotal = 251.00
```

Statements that mix int and double variables

```
int result9 = invoiceTotal / invoiceCount
                                      // result9 = 125
double result10 = invoiceTotal / invoiceCount
                                      // result10 = 125.50
```

The syntax for declaring and initializing a string variable

```
String variableName = value;
```

Statements that declare and initialize a string

```
String message1 = "Invalid data entry.";
String message2 = "";
String message3 = null;
```

How to join strings

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

How to join a string and a number

```
double price = 14.95;
String priceString = "Price: " + price;
```

How to append one string to another with the + operator

```
firstName = "Bob";      // firstName is Bob
lastName = "Smith";     // lastName is Smith
name = firstName + " ";
                        // name is Bob followed by a space
name = name + lastName;
                        // name is Bob Smith
```

How to append one string to another with the += operator

```
firstName = "Bob";      // firstName is Bob
lastName = "Smith";     // lastName is Smith
name = firstName + " ";
                        // name is Bob followed by a space
name += lastName;       // name is Bob Smith
```

Common escape sequences

- `\n`
- `\t`
- `\r`
- `\"`
- `\\`

A string with a new line

String

```
"Code: JSPS\nPrice: $49.50"
```

Result

```
Code: JSPS
Price: $49.50
```

A string with tabs and returns

String

```
"Joe\tSmith\rKate\tLewis"
```

Result

```
Joe       Smith
Kate      Lewis
```

A string with quotation marks

String

```
"Type \"x\" to exit"
```

Result

```
Type "x" to exit
```

A string with a backslash

String

```
"C:\\java\\files"
```

Result

```
C:\java\files
```

How to create an object from a class

Syntax

```
ClassName objectName = new ClassName(arguments);
```

Examples

```
Scanner sc = new Scanner(System.in);
                        // creates a Scanner object named sc
Date now = new Date();
                        // creates a Date object named now
```

How to call a method from an object

Syntax

```
objectName.methodName(arguments)
```

Examples

```
double subtotal = sc.nextDouble();
                        // get a double entry from the console
String currentDate = now.toString();
                        // convert the date to a string
```

How to call a static method from a class

Syntax

```
ClassName.methodName(arguments)
```

Examples

```
String sPrice = Double.toString(price);
                                // convert a double to a string
double total = Double.parseDouble(userEntry);
                                // convert a string to a double
```

Common packages

- `java.lang`
- `java.text`
- `java.util`
- `java.io`
- `java.sql`
- `java.applet`
- `java.awt`
- `java.awt.event`
- `javax.swing`

The syntax of the import statement

```
import packagename.ClassName;
   or
import packagename.*;
```

Examples

```
import java.text.NumberFormat;
import java.util.Scanner;
import java.util.*;
import javax.swing.*;
```

How to use the Scanner class to create an object

With an import statement

```
Scanner sc = new Scanner(System.in);
```

Without an import statement

```
java.util.Scanner sc = new java.util.Scanner(System.in);
```

Relational operators

Operator	Name
==	Equality
!=	Inequality
>	Greater Than
<	Less Than
>=	Greater Than Or Equal
<=	Less Than Or Equal

Examples of conditional expressions

```
discountPercent == 2.3  // equal to a numeric literal
subtotal != 0           // not equal to a numeric literal
years > 0               // greater than a numeric literal
i < months              // less than a numeric variable
subtotal >= 500
            // greater than or equal to a numeric literal
quantity <= reorderPoint
            // less than or equal to a numeric variable
```

Two methods of the String class

- `equals(String)`
- `equalsIgnoreCase(String)`

Examples

```
userEntry.equals("Y") // equal to a string literal
userEntry.equalsIgnoreCase("Y")
                      // equal to a string literal
(!lastName.equals("Jones"))
                      // not equal to a string literal
code.equalsIgnoreCase(productCode)
                      // equal to another string variable
```

The syntax of the if/else statement

```
if (booleanExpression) {statements}
[else if (booleanExpression) {statements}] ...
[else {statements}]
```

If statements without else if or else clauses

With a single statement

```
if (subtotal >= 100)
    discountPercent = .2;
```

With a block of statements

```
if (subtotal >= 100)
{
    discountPercent = .2;
    status = "Bulk rate";
}
```

An if statement with an else clause

```
if (subtotal >= 100)
    discountPercent = .2;
else
    discountPercent = .1;
```

An if statement with else if and else clauses

```
if (customerType.equals("T"))
    discountPercent = .4;
else if (customerType.equals("C"))
    discountPercent = .2;
else if (subtotal >= 100)
    discountPercent = .2;
else
    discountPercent = .1;
```

A loop that calculates the sum of the numbers 1 through 4

```
int i = 1;
int sum = 0;
while (i < 5)
{
    sum = sum + i;
    i = i + 1;
}
```