Java Programming
Fourth Edition

Chapter 1

Creating Your First Java Classes

Objectives

- Learn about programming
- Be introduced to object-oriented programming concepts
- Learn about Java
- Analyze a Java application that uses console output

Objectives (continued)

- Add comments to a Java class
- Save, compile, and run a Java application
- Modify a Java class
- Create a Java application using GUI output
- Correct errors and find help

Learning About Programming

- Program
  - Set of written instructions which tells computer what to do
- Machine language
  - Most basic circuitry-level language
  - Problems
    - Keep track of many switches
    - Hard to discover errant switches

Learning About Programming (continued)

- High-level programming language
  - Allows you to use vocabulary of reasonable terms
- Syntax
  - Rules of language
- Program statements
  - Similar to English sentences
  - Carry out tasks of program

Learning About Programming (continued)

- Compiler or interpreter
  - Translates language statements into machine code
- Syntax error
  - Misuse of language
  - Misspelled programming language word
- Debugging
  - Freeing program of all errors
- Logic errors
  - Also called semantic errors
  - Incorrect order or procedure

Introducing Object-Oriented Programming Concepts

- Procedural programming
  - Sets of operations executed in sequence
  - Create names for computer memory locations that hold values
- Procedures
  - Individual operations grouped into logical units
- Object-oriented programs
  - Create classes
  - Create objects from classes
  - Create applications

Introducing Object-Oriented Programming Concepts (continued)

- GUI system
  - Allows you to treat files as objects
  - Not all object-orientated programs written to use GUI
- Object-oriented programming differs from traditional procedural programming
  - Basic concepts
    - Objects
    - Classes
    - Inheritance
    - Polymorphism

Understanding Objects And Classes

- Objects
  - Made up of attributes and methods
  - Attributes called states
- Attributes
  - Characteristics that define object
  - Differentiate objects of same class
- Class
  - Describes group or collection of objects with common properties

Understanding Objects And Classes (continued)

- Method
  - Self-contained block of program code
  - Similar to procedure
- Encapsulation
  - Refers to hiding of data and methods within object
  - Provides security
  - Keeps data and methods safe from inadvertent changes

Understanding Inheritance and Polymorphism

- Inheritance
  - Important feature of object-oriented programs
  - Classes share attributes and methods of existing classes but have more specific features
  - Helps you understand real-world objects
- Polymorphism
  - Means “many forms”
  - Allows same word to be interpreted correctly in different situations based on context

Learning About Java

- Java
  - Developed by Sun Microsystems
  - Object-oriented language
  - General-purpose
  - Advantages
    - Security features
    - Architecturally neutral

Learning About Java (continued)

- Java (continued)
  - Can be run on wide variety of computers
  - Does not execute instructions on computer directly
  - Runs on hypothetical computer known as Java virtual machine (JVM)
- Source code
  - Programming statements written in high-level programming language

Learning About Java (continued)

- Bytecode
  - Statements saved in file
  - Java compiler converts source code into binary program
- Java interpreter
  - Checks bytecode and communicates with operating system
  - Executes bytecode instructions line-by-line within Java virtual machine

The Java Environment

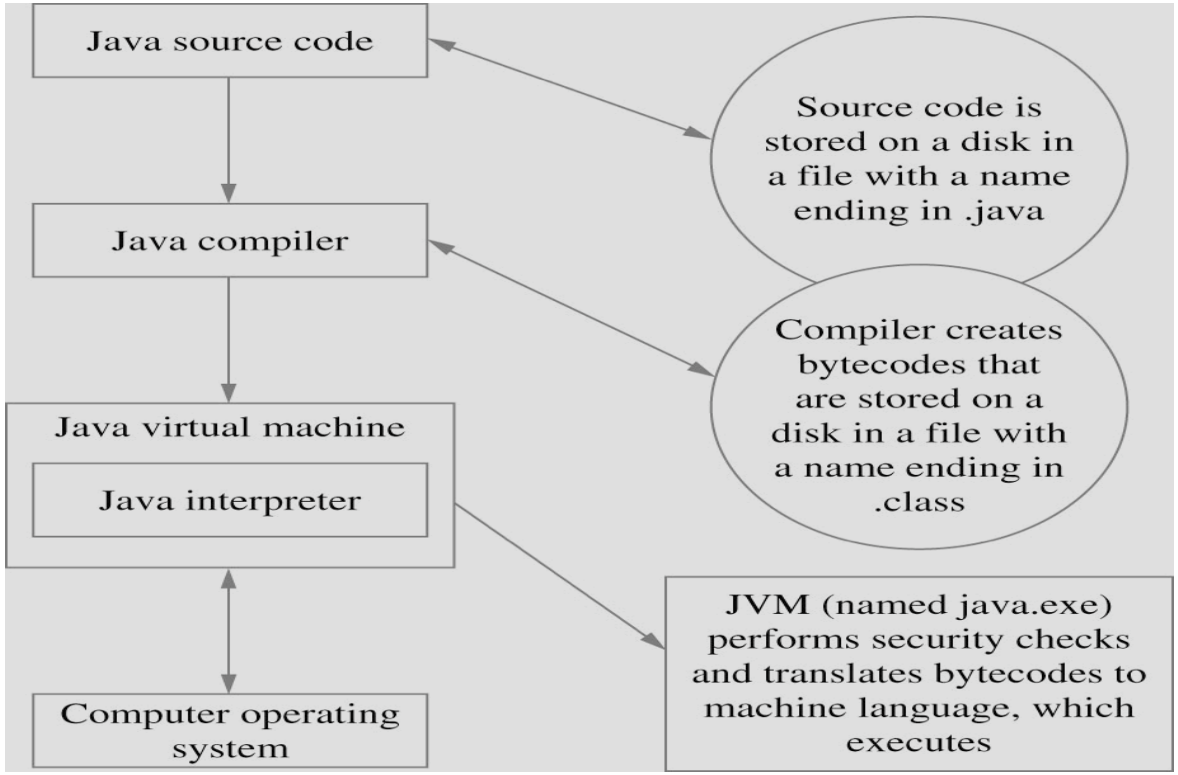

Figure 1-1 The Java environment

Java Program Types

- Applets
  - Programs embedded in Web page
- Java applications
  - Called Java stand-alone programs
  - Console applications
    - Support character output
  - Windowed applications
    - Menus
    - Toolbars
    - Dialog boxes

Analyzing a Java Application That Uses Console Output

- Even simplest Java application
  - Involves fair amount of confusing syntax
- Print “First Java application” on screen

The `First` Class

```
public class First
{
    public static void main(String[] args)
    {
        System.out.println("First Java application");
    }
}
```

Figure 1-2 The `First` class

Understanding the Statement That Prints the Output

- Literal string
  - Will appear in output exactly as entered
  - Written between double quotation marks
- Arguments
  - Pieces of information passed to method
- Method
  - Requires information to perform its task

Understanding the `First` Class

- Everything used within Java program must be part of a class
- Define Java class using any name or identifier
- Requirements for identifiers
  - Must begin with:
    - Letter of English alphabet
    - Or non-English letter (such as α or π)
  - Cannot begin with digit

Understanding the `First` Class (continued)

- Requirements for identifiers
  - Can only contain:
    - Letters
    - Digits
    - Underscores
    - Dollar signs
  - Cannot be Java reserved keyword
  - Cannot be true, false, or null
- Access modifier
  - Defines how class can be accessed

Java Reserved Keywords

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

Table 1-1 Java reserved keywords

Legal but Unconventional and Nonrecommended Class Names in Java

Class Name	Description
`Undergradstudent`	New words are not indicated with initial uppercase letters; difficult to read
`Inventory_Item`	Underscore is not commonly used to indicate new words
`BUDGET2009`	Using all uppercase letters is not common

Table 1-3 Legal but unconventional and nonrecommended class names in Java

Some Illegal Class Names in Java

Class Name	Description
`an employee`	Space character is illegal
`Inventory Item`	Space character is illegal
`class`	`class` is a reserved word
`2009Budget`	Class names cannot begin with a digit
`phone#`	# symbol is illegal

Table 1-4 Some illegal class names in Java

Understanding the `main()` Method

- `Static`
  - Reserved keyword
  - Means method accessible and usable
    - Even though no objects of class exist
- `void`
  - use in `main()` method header
  - Does not indicate `main()` method empty
  - Indicates `main()` method does not return value when called
  - Doesn't mean `main()` doesn't produce output

Shell Code

```
public class AnyClassName
{
    public static void main(String[] args)
    {
        /******/
    }
}
```

Figure 1-3 Shell code

Adding Comments to a Java Class

- Program comments
  - Non-executing statements added to program for documentation
  - use to leave notes for yourself or others
  - Include author, date, class's name or function

Adding Comments to a Java Class (continued)

- Types of Java comments
  - Line comments
    - Start with two forward slashes (//)
    - Continue to end of current line
    - Do not require ending symbol
  - Block comments
    - Start with forward slash and asterisk (/*)
    - End with asterisk and forward slash (*/)

Adding Comments to a Java Class (continued)

- Types of Java comments (continued)
  - Javadoc comments
    - Special case of block comments
    - Begin with slash and two asterisks (/**)
    - End with asterisk and forward slash (*/)
    - use to generate documentation

A Program Segment Containing Several Comments

```
// Demonstrating comments
/* This shows
    that these comments
   don't matter */
System.out.println("Hello"); // This line executes
   // up to where the comment started
/* Everything but the println()
     is a comment */
```

Figure 1-4 A program segment containing several comments

Saving, Compiling, and Running a Java Application

- Use Java interpreter to:
  - Translate bytecode into executable statements
- Reasons for error messages
  - Misspelled command javac
  - Misspelled filename
  - Not within correct subfolder or subdirectory on command line
  - Java not installed properly

Running a Java Application

- Run application from command line
- Shows application's output in command window
- Class stored in folder named Java on C drive

Output of the `First` Application

Figure 1-5 Output of the `First` application

Modifying a Java Class

- To execute new source code
  - Save file with changes (using same filename)
  - Compile class with `javac` command
  - Interpret class bytecode
  - Execute class using `java` command

Creating a Java Application Using GUI Output

- `JOptionPane`
  - Produce dialog boxes
- Dialog box
  - GUI object resembling window
  - Messages placed for display
- Package
  - Group of classes
- `import` statement
  - Use to access built-in Java class

Output of the `FirstDialog` Application

Figure 1-8 Output of the `FirstDialog` application

Correcting Errors and Finding Help

- First line of error message displays:
  - Name of file where error found
  - Line number
  - Nature of error
- Next lines identify:
  - Symbol
  - Location
- If compiler detects violation of language rules
  - Refuses to translate class to machine code

You Do It

- Your first application
- Adding comments to a class
- Modifying a class
- Creating a dialog box

Summary

- Computer program
  - Set of instructions that tells a computer what to do
- Object-oriented programs
  - Classes
  - Objects
  - Applications
- Java virtual machine (JVM)
  - Standardized hypothetical computer
- Everything in a Java program must be part of a class

Summary (continued)

- Access modifier
  - Word that defines circumstances under which class can be accessed
- All Java applications must have method named `main()`
- Program comments
  - Nonexecuting statements
  - Add to file for documentation
- `javac`
  - Compile command

Summary (continued)

- `java`
  - Execute command
- `JOptionPane`
  - GUI
  - Provides methods for creating dialogs