Chapter 1

Introduction to Java

Lesson page 0-1. Introduction to Livetexts

Question 1. A livetext is a text that relies not only on the printed word but also on graphics, animation, audio, the computer, and the active participation of the "reader".

Lesson page 1-1. Hardware and software

Question 1. A program is a set of instructions, written in a programming language, for a computer to execute.

Question 2. A programming language is a language in which programs are written for a computer to carry out or execute.

Question 3. The machine language is the language that a computer is designed to execute.

Question 4. An assembly language is a symbolic language that reflects the machine language of a computer —it includes one symbolic instruction for each machine language instruction.

Activity 1-1-1 Hardware versus software

Activity 1-1-2 Computer memory

Question 5. Memry is the set of locations (bytes) of a computer that holds the programs and the data that the programs are manipulating when the program is being executed. Each location has a number by which it is referenced, called its address. Typically, the contents of memory is lost when the computer is turned off.

Question 6. The address of a memory location is the number by which a memory location is referenced. The first memory location has address 0, the second 1, and so on.

Question 7. A bit is either a 0 or a 1. Short for "binary digit".

Question 8. A byte is an eight-bit quantity, a memory location consisting of eight bits, and also a Java primitive type.

Question 9. Symbols used in decimal notation: `0 1 2 3 4 5 6 7 8 9`.

Question 10. Symbols used in binary notation: `0 1`.

Question 11. Here are the binary equivalents of the first 20 decimal numbers:

0:	0	5:	101	10:	1010	15:	1111
1:	1	6:	110	11:	1011	16:	10000
2:	10	7:	111	12:	1100	17:	10001
3:	11	8:	1000	13:	1101	18:	10010
4:	100	9:	1001	14:	1110	19:	10011

Lesson page 1-2. Some simple Java programs

Activity 1-2-1 A do-nothing Java program

Question 1. In Java, a keyword is a word that has been reserved for a special purpose.

Question 2. A method is a parameterized sequence of instructions. Java has three kinds of method: procedure, function, and constructor.

Question 3. Execution starts by calling method `main`.

Question 4. Every program has a method called `main`.

Question 5. The class is the basic building block of Java.

Question 6. The body of a class begins with { and ends with }.

Question 7. True; instruction and statement mean the same thing.

Question 8. Here's the program with incorrect parts boxed:

```
// This program won't even compile.
public class Garble (
    public static void start(String[] args) {
    }
)
```

Activity 1-2-2 A program that prints

Question 9. The Java console is a window that appears when a Java program is executed. Output from `println` statements will appear here.

Question 10. The instruction `println` moves the cursor to the next line after printing its argument in the Java console; `print` does not.

Question 11. The following appears on the Java console, and the cursor is right after the `e` of `everyone`.

```
Hello, everyone
```

Question 12. Here's the program:

```
public class PrintPractice {
    public static void main(String[] arguments) {
        System.out.println("Frances");
        System.out.println();
        System.out.println("#5 from the lighthouse");
    }
}
```

Activity 1-2-3 A program that creates a window

Question 13. API stands for Application Programming Interface.

Question 14. GUI stands for Graphical User Interface.

Question 15. A package is a collection of classes.

Question 16. Package `java.io` deals with input/output.

Question 17. Package `javax.swing` deals with GUI components.

Question 18. This must appear at before the class: **import** `javax.swing.*;`

Question 19. Class `JFrame` deals with windows.

Question 20. Here's the program:

```
import javax.swing.*;

public class WindowPractice {
    public static void main(String[] args) {
        (new JFrame()).show();
        (new JFrame()).show();
        (new JFrame()).show();
    }
}
```

Activity 1-2-4 A program that draws

Question 21. **import** `javax.swing.*; and` **import** `java.awt.*;`

Question 22. Method `paint` redraws the graphics window:

Question 23. Method `paint` is called by the system when it decides that the graphics window has to be redrawn (e.g. if you had covered it with another window and then uncovered it).

Question 24. Originally, the origin (pixel `(0,0)`) is the top-left corner of the window. In this program, it is has been changed to be under the title.

Question 25. This statement sets the color: `g.setColor(Color.black);`

Question 26. In `(a,b)`, `a` is the horizontal coordinate.

Question 27. False; the two pixels are the same distance from the bottom.

Question 28. False.

Question 29. `g.drawString("Wahoo!", 40, 60);`

Question 30. Here's the modified program:

```
// A simple graphics window.
import java.awt.*;

public class JLiveGraphics1 extends Frame {
    // Repaint the Graphics Window with a silly face.
    public void paint(Graphics g) {
        g.translate(getInsets().left, getInsets().top);
        g.setColor(Color.black);
        g.drawRect(77,10,40,30);
        g.drawString("rectangle",77,60);
        g.setColor(Color.red);
        g.drawOval(15,10,30,30);
        g.drawString("circle",15,60);

        g.setColor(Color.black);
        g.drawOval(15,10,90,90);
        g.setColor(Color.blue);
        g.drawOval(40,34,10,10);
        g.drawOval(70,34,10,10);
        g.setColor(Color.red);
        g.drawRect(58,50,4,15);
        g.drawOval(40,75,40,5);
    }
}
```

Activity 1-2-5 Simple Java program exercises

Lesson page 1-3. Components of a Java program

Activity 1-3-1 Comments and whitespace

Question 1. Comments improve the reader's ability to understand the program.

Question 2. The comment begins with `/*` and ends with `*/`.

Question 3. `// Store in s the sum of the integers from 1 to n`

Question 4. The five forms of whitespace are given below. Whether you get an end-of-line character, a return character, or both when you press the return/enter key depends on the operating system you ar using.

1. The space character, which you get by typing the space bar on the keyboard.

2. The tab character, which you get by typing the tab key.
3. The end-of-line character, you may get by typing the return/enter key.
4. The return character, which you may get by typing the return/enter key.
5. A comment.

Question 5. False; comments are a form of whitespace.

Question 6. Whitespace can appear between symbols of a Java program.

Question 7. Whitespace cannot appear in the middle of a symbol.

Question 8. Here's the program:

```java
// A simple graphics window.
import java.awt.*;

public class JLiveGraphics1 extends Frame {

    /* Repaint the Graphics Window with a silly face.
     */
    public void paint(Graphics g) {
        /* g.translate(getInsets().left, getInsets().top);
        g.setColor(Color.black);
        g.drawRect(77,10,40,30);
        g.drawString("rectangle",77,60);
        g.setColor(Color.red);
        g.drawOval(15,10,30,30);
        g.drawString("circle",15,60);
        */

        // Draw the circular face, two blue eyes,
        // a narrow rectangular nose, and oval mouth.
            g.setColor(Color.black);
            g.drawOval(20,65,80,80);
            g.setColor(Color.blue);
            g.drawOval(40,84,10,10);
            g.drawOval(70,84,10,10);
            g.setColor(Color.red);
            g.drawRect(58,100,4,15);
            g.drawOval(40,125,40,5);
    }
}
```

Activity 1-3-2 Exercises on comments and whitespace

Activity 1-3-3 Keywords and identifiers

Question 9. In Java, a keyword is a word that has been reserved for a special purpose.

Question 10. **import, public** (used twice)**, class, static, void.**

Question 11. An identitfier is a case-sensitive sequence of characters where:

1. The first character is a lowercase letter, an uppercase letter, an underscore, or a dollar sign.
2. Each other character is a digit, a lowercase letter, an uppercase letter, an underscore, or a dollar sign.
3. Keywords may not be used as identifiers.
4. It is advisable not to use the dollar sign in identifiers because identifiers with dollar signs are used in special circumstances, which may conflict with your use of them.

Question 12. Another word for *alphameric* is *alphanumeric*.

Question 13. The dollar sign $ and the underscore _ can appear in identifiers.

Question 14. False. Identifiers are case-sensitive in Java.

Question 15.
The conflicting guidelines are:

1. Long identifiers that relate to what they are used for are good.
2. To keep a program short, use short identifiers.

Activity 1-3-4 Exercises on keywords and identifiers

Activity 1-3-5 Variables and types

Question 16. A variable is a name together with an associated value.

Question 17. False. Keywords can not be used as identifiers.

Question 18. A type is a set of values together with operations on them.

Question 19. A variable declaration is a Java construct that declares a variable and gives its type. Its basic form is <*type*> <*identifier*> ;.

Question 20. The three parts of a variable declaration are:

1. a type (such as **int**), which describes the values the variable can contain;
2. an identifier, which is the name given to the variable; and
3. a semicolon.

Question 21. The range of **int** is: -2147483648..2147483647.

Question 22. The range of **byte** is: -128..127.

Question 23. Type String is not a primitive type; it is defined in package java.lang.

Activity 1-3-6 Expressions

Question 24. Division with both operands of type **int** is done as follows: perform conventional division; then truncate the result toward 0 to an integer. For example:

```
7 / 2 = 3
(-7) / 2 = -3
```

Question 25. Here's the program and its output:

```
public class IntPrint {
    public static void main(String[] bob) {
        System.out.println(5 + 2 * 3);
        System.out.println(5 + (2 * 3));
        System.out.println((5 + 2) * 3);
        System.out.println(5 / 2);
        System.out.println(7 / 2 * 4);
        System.out.println(7 / (3 * - 2));
    }
}
```

Output:

```
11
11
21
2
12
-1
```

Question 26. Here's a program that does the job:

```
public class IntPrint {
    public static void main(String[] bob) {
        int x= 5; int y= 2; int z= 3;
        z= 4;
        System.out.println(z);

        x= 5; y= 2; z= 3;
        z= y;
        System.out.println(z);

        x= 5; y= 2; z= 3;
    z= 22 / x;
    System.out.println(z);

    x= 5; y= 2; z= 3;
    z= x + z;
    System.out.println(z);

    x= 5; y= 2; z= 3;
    z= x + (z * y);
    System.out.println(z);

    x= 5; y= 2; z= 3;
    z= y + z * x;
    System.out.println(z);

    x= 5; y= 2; z= 3;
    z= (3 * (- (z)));
    System.out.println(z);
  }
}
```

Output:

```
4
2
4
8
11
17
-9
```

Activity 1-3-7 Exercises on variables and expressions

Lesson page 1-4. Three statements: assignment, conditional and block statements

Activity 1-4-1 The assignment statement

Question 1. <*variable*>= <*expression*>;

Question 2. <*type*> <*identifier*>;

Question 3. To execute an assignment statement: (1) evaluate the expression and (2) store the value of the expression in the variable.

Question 4. False. A variable may be assigned many times during execution of a program.

Question 5. Here's the program:

```
public class AssignmentPractice {
    public static void main(String[] args) {
        int cSquared= (3*3) + (5*5);
        System.out.print(cSquared);
    }
}
```

Output:

```
34
```

Activity 1-4-2 Swapping two variables

Question 6.

```
public class Swap {
    public static void main(String[] args) {
        int a= 1; int b= 2; int c= 3;

        // Rotate (the values of) a, b, and c.
            int temp= a;
            a= b;
            b= c;
            c= temp;
    }
}
```

Activity 1-4-3 Conditional statements: `if`

Question 7. The form of the if-statement is:

```
if ( <boolean expression> )
    <statement>
```

Question 8. A boolean expression evaluates to either **true** or **false**.

Question 9. If the condition is true, the statement is executed.

Question 10. If the condition is false, the if-statement terminates (without executing the statement).

Activity 1-4-4 Conditional statements: `if-else`

Question 11. The form of the if-else-statement is:

```
if ( <boolean expression> )
    <statement1>
else
    <statement2>
```

Question 12. If the condition is true, the then-part (*statement1*) is executed.

Question 13. If the condition is false, the else-part (*statement2*) is executed.

Activity 1-4-5 Boolean expressions

Question 14. The boolean values are: **true** and **false**.

Question 15. A boolean expression is an expression that evaluates to either **false** or **true**.

Question 16. A proposition is a declarative sentence; one that can be interpreted to be either true or false; it is verbal expression that is either false or true.

Question 17. These six expressions produce a boolean value (when evaluated):

1. a < b
2. a == b
3. a > b
4. a <= b
5. a != b
6. a >= b

Question 18. Three boolean operators are:

1. negation: !
2. and: &&
3. or: ||

Question 19. Here's the program:

```
public class BooleanPractice {
    public static void main(String[] arg) {
        System.out.println(true && false);
        System.out.println(true || false);
        System.out.println(false || false);
        System.out.println(false && false);
        System.out.println(false && true);
        System.out.println(false || true);
        System.out.println(false && !true);
        System.out.println(!false);
        System.out.println(!true);
    }
}
```

Output:

```
false
true
false
false
false
true
false
true
false
```

Activity 1-4-6 Blocks

Question 20. To aggregate means to bring together, to collect into one sum, mass, or body. Accordingly, an aggregate is a sum, mass, or assemblage of particulars.

Question 21. The block has the form

{ <*sequence of statements and declarations*> }

Question 22. False.

Question 23. The rule is: indent substatements of a block to help make the structure of a program clear.

Question 24. Here is the fixed program:

```
public class ErrorRiddled {
    public static void main(String[] args) {
    }
}
```

Activity 1-4-7 Exercises on assignments, conditionals, and blocks

Question 25. First version:

```
public class IfX {
    public static void main(String args[]) {
        int y= 3;
        int x= 4;
        if (x < y)
            x= y + x;
        System.out.println("Now x >= y.");
    }
}
```

Second version:

```
public class IfX {
    public static void main(String args[]) {
        int y= 3;
        int x= 2;
        if (x < y) {
            x= y + x;
            System.out.println("Now x < y.");
        }
    }
}
```

Lesson page 1-5. Input/Output

Activity 1-5-1 Using the Java Console for output

Question 1. The output of the program is:

```
Once more unto the breach, dear friends, once more;
Or close the wall up...
-Shakespeare, King Henry V, Act 3, Scene 1
```

Activity 1-5-2 Using the Java Console for input

Question 2. To prompt means to give a cue, to assist by suggesting something. In computing, one prompts the user to input text from the keyboard (or do other things as well) by printing a message of some form.

Question 3. Add a space after the `"i"` in `"Hi"` or before the `"P"` in `"Phred"`.

```
high 5
340
5.0 is five        See note 1 below.
Is five 3.41.6     See note 2 below.
```

1. The values (from left to right) are of type `double, double, String`. The first "+" the second results in a `double`. Anything "+" a `String` results in a `String`. So `3.4 + 1.6 = 5.0; 5.0 + " is five"` results in `"5.0 is five"`.

2. The values (from left to right) are: `String, double, double`. A `String` ”+” anything results in a `String`. So `"Is five " + 3.4` results in `"Is five 3.4"`. Then `"Is five 3.4" + 1.6` results in `"Is five 3.41.6"`.

Activity 1-5-3 A GUI: JLiveWindow

Question 4.

1. `MyJLiveWindow testJLiveWindow= new MyJLiveWindow(3, 0, 0);`

2. (a) Multiply the number in field 0 by the number in field 1 and store the result in `mult`.
   `setIntField(2,mult);`

   (b) Subtract the number in field 1 from the number in field 0 and store the result in `other`.
   `setIntField(2,other);`

   (c) Divide the number in field 0 by the number in field 1 and store the result in `divResult`.
   `setIntField(2,divResult);`

If you type 4 into field 0 and 3 into field 1, 1 is displayed in field 2. The division results in an integer, and the fraction part is discarded.

Activity 1-5-4 Exercises on the Java console and GUIs

Activity 1-5-5 Graphics windows

Question 5. No answer is provided.

Activity 1-5-6 Using a graphics window

Question 6. We would expect this statement to store the larger of `a` and `b` in `x`.

Question 7. Here's method `paint`:

```
public void paint(Graphics g) {
    g.translate(getInsets( ).left, getInsets( ).top);
    g.drawOval(20,20,40,40);
    g.setColor(Color.magenta);
    g.drawOval(40,20,40,40);
    g.setColor(Color.blue);
    g.drawOval(30,40,40,40);
}
```

Activity 1-5-7 Exercises on graphics windows

Lesson page 1-6. Assertions in programs

Activity 1-6-1 Relations about variables and values

Question 1. A relation is a true-false statement.

Question 2. In Java, the symbol for mathematical equality is: `==`

Activity 1-6-2 Simplifying a particular kind of relation

Question 3. True. *Dogs* can mean *feet.*

Question 4. If it is not raining, `wetPeople` could be true or false.

Question 5. If it is raining, `wetPeople` is true; otherwise `wetPeople` is false. Or: "It is raining" = `wetPeople`.

Question 6. `greater0 == (x > 0)`

Question 7. I'll eat my hat = you're right.
Or: If you're right, I'll eat my hat; otherwise I won't.

Question 8. `rain=` "it is raining"

Activity 1-6-3 Assertions

Question 9. An assertion is a relation that is placed in a program (as a comment) in order to assert that it is true when execution of the program reaches that place.

Question 10. A Hoare triple consists of an assertion (the precondition) followed by a statement followed by another assertion (the postcondition). It has the meaning: execution of the statement begun with the precondition true is guaranteed to terminate, and when it terminates, the postcondition will be true.

Question 11. A precondition of a statement is an assertion that is placed before the statement.

Question 12. A postcondition of a statement is an assertion that is placed after the statement.

Question 13. This is what the code means: if `(x+y)*y = 82` is true before execution of the statement `x= x+y;`, then execution will terminate, and when it does, `x*y = 82` will be true.

Lesson page 1-7. Additional statements and expressions