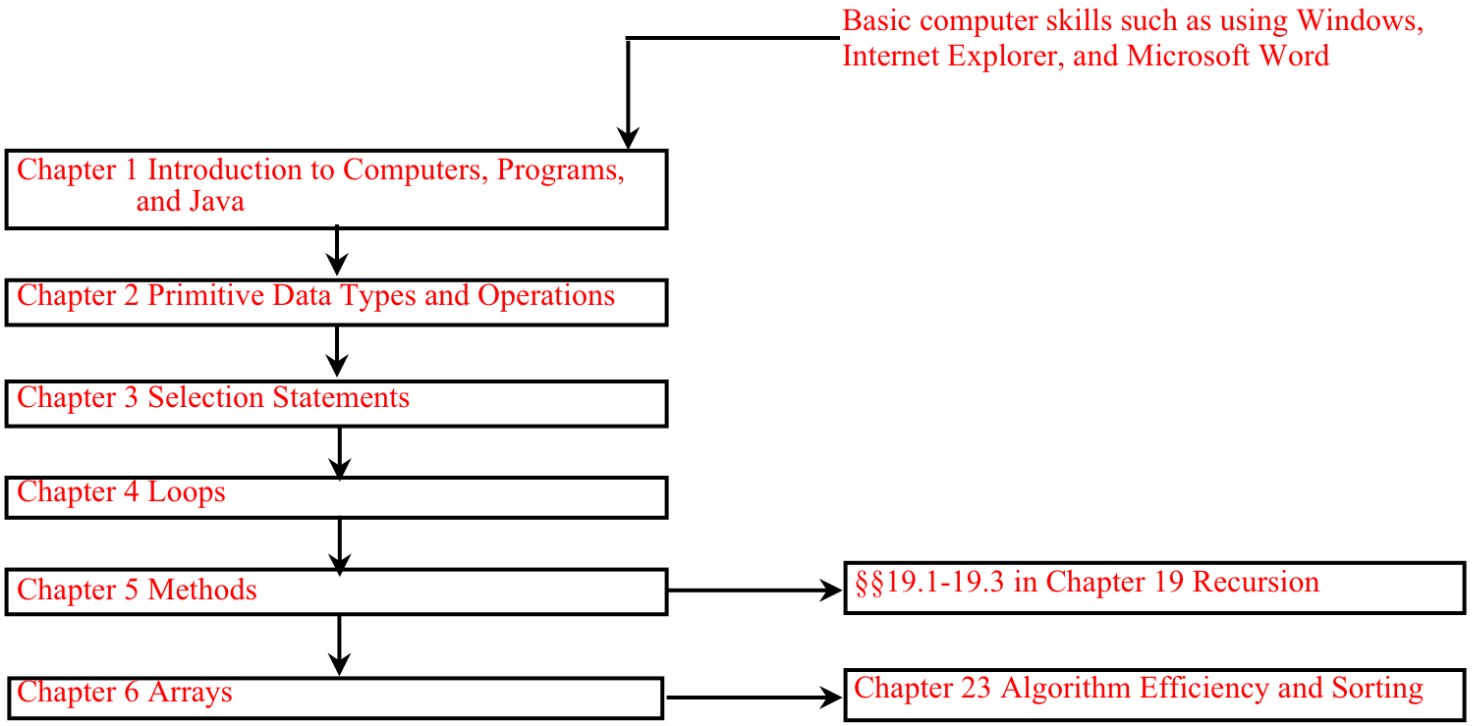
Chapter 4 Loops



Objectives

- To use while, do-while, and for loop statements to control the repetition of statements (§§4.2-4.4).
- To understand the flow of control in loop statements (§§4.2-4.4).
- To use Boolean expressions to control loop statements (§§4.2-4.4).
- To write nested loops (§4.5).
- To know the similarities and differences of three types of loops (§4.6).
- To implement program control with break and continue (§4.7).

Note

The initial-action in a for loop can be a list of zero or more comma-separated expressions. The action-after-each-iteration in a for loop can be a list of zero or more comma-separated statements. Therefore, the following two for loops are correct. They are rarely used in practice, however.

```
for (int i = 1; i < 100; System.out.println(i++));


for (int i = 0, j = 0; (i + j < 10); i++, j++) {
  // Do something
}
```

Note

If the loop-continuation-condition in a for loop is omitted, it is implicitly true. Thus the statement given below in (a), which is an infinite loop, is correct. Nevertheless, it is better to use the equivalent loop in (b) to avoid confusion:

```
for ( ; ; ) {
  // Do something
}
```

(a)

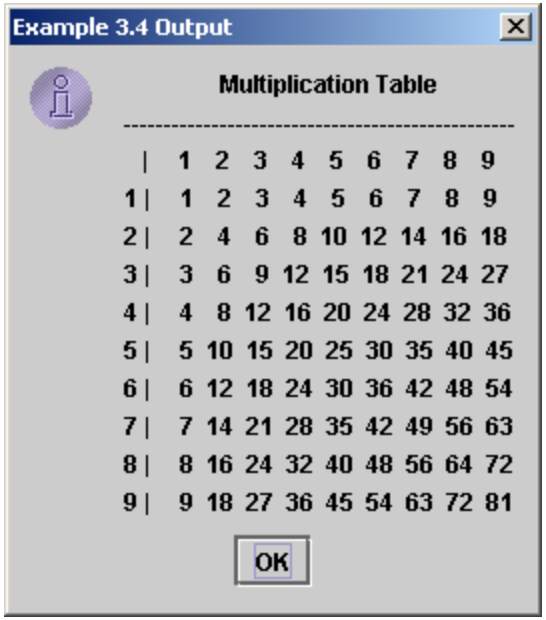
Equivalent

```
while (true) {
  // Do something
}
```

(b)

Nested Loops

Problem: Write a program that uses nested for loops to print a multiplication table.



TestMultiplicationTable

Caution

Adding a semicolon at the end of the for clause before the loop body is a common mistake, as shown below:

Logic Error

```
for (int i=0; i<10; i++);
{
  System.out.println("i is " + i);
}
```