Flow of Control: Loops

Chapter 4
Java Loop Statements: Outline

• The **while** statement
• The **do-while** statement
• The **for** Statement
Java Loop Statements

• A portion of a program that repeats a statement or a group of statements is called a *loop*.

• The statement or group of statements to be repeated is called the *body* of the loop.

• There must be a means of exiting the loop.
The **while** Statement

- Also called a **while** loop
- A **while** statement repeats while a controlling boolean expression remains true
- The loop body typically contains an action that ultimately causes the controlling boolean expression to become false.
The **while** Statement

- View sample program, Listing 4.1

```java
class WhileDemo {
    public static void main(String[] args) {
        int number = -1;
        String message = "Buckle my shoe."
        do {
            System.out.println(message);
            number = Integer.parseInt(System.in.readLine());
        } while (number != 0);
    }
}
```

Sample screen output:

```
Enter a number:
2
1, 2,
Buckle my shoe.

Enter a number:
3
1, 2, 3,
Buckle my shoe.

Enter a number:
0
```

*The loop body is iterated zero times.*
The **while** Statement

• Syntax

```java
while (Boolean_Expression)
    Body_Statement

or

while (Boolean_Expression)
{
    First_Statement
    Second_Statement
    ...
}
```
The **while** Statement

- Figure 4.2

Semantics of the **while** statement
The **while** Statement

- Figure 4.1
  The action of the **while** loop in Listing 4.1
The **do-while** Statement

- Also called a **do-while** loop
- Similar to a **while** statement, except that the loop body is executed at least once
- Syntax
  
  ```
  do
  Body_Statement
  while (Boolean_Expression);
  ```

- Don’t forget the semicolon!
The **do-while** Statement

- View sample program, listing 4.2

```java
class DoWhileDemo

Enter a number:
2
1, 2,
Buckle my shoe.

Enter a number:
3
1, 2, 3,
Buckle my shoe.

Enter a number:
0
1,
Buckle my shoe.

The loop body always executes at least once.
```

Sample screen output
The **do-while** Statement

- Figure 4.3 The Action of the **do-while** Loop in Listing 4.2

```java
do
{
    System.out.print(count + "", "");
    count++;
} while (count <= number);
```
The **do-while** Statement

• First, the loop body is executed.
• Then the boolean expression is checked.
  ▪ As long as it is true, the loop is executed again.
  ▪ If it is false, the loop is exited.
• Equivalent **while** statement

```
Statement(s)_S1
while (Boolean_Condition)
  Statement(s)_S1
```
The **do-while** Statement

- Figure 4.4 The Semantics of the **do-while** Statement
Programming Example: Bug Infestation

• Given
  ▪ Volume a roach: 0.002 cubic feet
  ▪ Starting roach population
  ▪ Rate of increase: 95%/week
  ▪ Volume of a house

• Find
  ▪ Number of weeks to exceed the capacity of the house
  ▪ Number and volume of roaches
Programming Example: Bug Infestation

Algorithm for roach population program (rough draft)

2. Get initial number of roaches in house.
3. Compute number of weeks until the house is full of roaches.
4. Display results.
Programming Example: Bug Infestation

Variables Needed

GROWTH_RATE — weekly growth rate of the roach population (a constant 0.95)

ONE_BUG_VOLUME — volume of an average roach (a constant 0.002)

houseVolume — volume of the house

startPopulation — initial number of roaches

cont. ...
Programming Example: Bug Infestation

Variables Needed

**countWeeks** — week counter

**Population** — current number of roaches

**totalBugVolume** — total volume of all the roaches

**newBugs** — number of roaches hatched this week

**newBugVolume** — volume of new roaches
Programming Example: Bug Infestation

• View sample program, listing 4.3

class BugTrouble

Enter the total volume of your house in cubic feet: 20000
Enter the estimated number of roaches in your house: 100
Starting with a roach population of 100 and a house with a volume of 20000.0 cubic feet, after 18 weeks, the house will be filled with 16619693 roaches. They will fill a volume of 33239 cubic feet. Better call Debugging Experts Inc.
Infinite Loops

• A loop which repeats without ever ending is called an infinite loop.

• If the controlling boolean expression never becomes false, a while loop or a do-while loop will repeat without ending.

• A negative growth rate in the preceding problem would cause totalBugVolume to always be less than houseVolume, so that the loop never ends.
Nested Loops

• The body of a loop can contain any kind of statements, including another loop.
Nested Loops

• View sample program, listing 4.4

class ExamAverager

Sample screen output

Want to average another exam? 
Enter yes or no. 
yes

Enter all the scores to be averaged. 
Enter a negative number after 
you have entered all the scores. 
90
70
80
−1
The average is 80.0
Want to average another exam? 
Enter yes or no. 
no
Nested Loops

• In the previous example
  ▪ The average score was computed using a while loop.
  ▪ This while loop was placed inside a do-while loop so the process could be repeated for other sets of exam scores.