

Chapter Six - Iteration

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Chapter Goals

- To be able to program loops with the `while` and `for` statements
 - To avoid infinite loops and off-by-one errors
 - To be able to use common loop algorithms
 - To understand nested loops
 - To implement simulations
- T** To learn about the debugger

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

while Loops

- A `while` statement executes a block of code repeatedly
- A condition controls how often the loop is executed

```
while (condition)
    statement
```

- Most commonly, the statement is a block statement (set of statements delimited by `{ }`)

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Calculating the Growth of an Investment

- Want to know when has the bank account reached a particular balance:

```
while (balance < targetBalance)
{
    years++;
    double interest = balance * rate / 100;
    balance = balance + interest;
}
```

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Execution of a while Loop

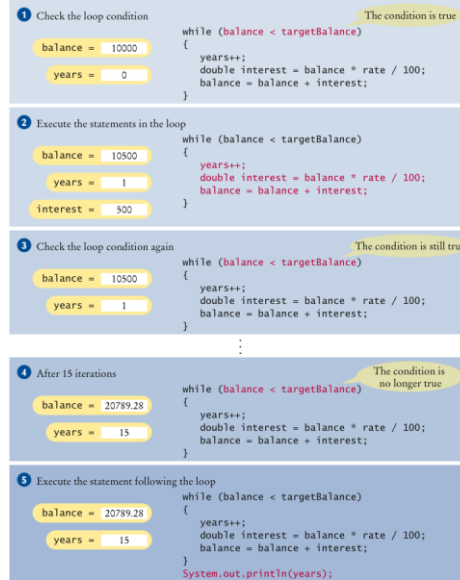


Figure 1 Execution of a while Loop

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Syntax 6.1 The while Statement

Syntax `while (condition)`
`statement`

Example

```

double balance = 0;
.
.
while (balance < TARGET)
{
    double interest = balance * RATE / 100;
    balance = balance + interest;
}
    
```

This variable is declared outside the loop and updated in the loop.

If the condition never becomes false, an infinite loop occurs.

This variable is created in each loop iteration.

Beware of "off-by-one" errors in the loop condition.

Don't put a semicolon here!

These statements are executed while the condition is true.

Lining up braces is a good idea.

Braces are not required if the body contains a single statement, but it's good to always use them.

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Self Check 6.1

How often is the following statement in the loop executed?

```
while (false) statement;
```

Answer: Never.

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Common Error: Infinite Loops

- Example:

```
int years = 0;
while (years < 20)
{
    double interest = balance * rate / 100;
    balance = balance + interest;
}
```

- Loop runs forever — must kill program

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Common Error: Infinite Loops

- Example:

```
int years = 20;
while (years > 0)
{
    years++; // Oops, should have been years--
    double interest = balance * rate / 100;
    balance = balance + interest;
}
```

- Loop runs forever — must kill program

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Common Error: Off-by-One Errors

- **Off-by-one error:** a loop executes one too few, or one too many, times
- Example:

```
int years = 0;
while (balance < 2 * initialBalance)
{
    years++;
    double interest = balance * rate / 100;
    balance = balance + interest;
}
System.out.println("The investment reached the target after " +
years + " years.");
```

- Should `years` start at 0 or 1?
- Should the test be `<` or `<=`?

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Avoiding Off-by-One Error

- Look at a scenario with simple values:
 initial `balance`: \$100
 interest `rate`: 50%
 after year 1, the `balance` is \$150
 after year 2 it is \$225, or over \$200
 so the investment doubled after 2 years
 the loop executed two times, incrementing `years` each time
Therefore: `years` must start at 0, not at 1.
- interest `rate`: 100%
 after one year: `balance` is $2 * \text{initialBalance}$
 loop should stop
Therefore: must use `<`
- Think, don't compile and try at random

Big Java by Cay Horstmann
 Copyright © 2009 by John Wiley & Sons. All rights reserved.

for Loops

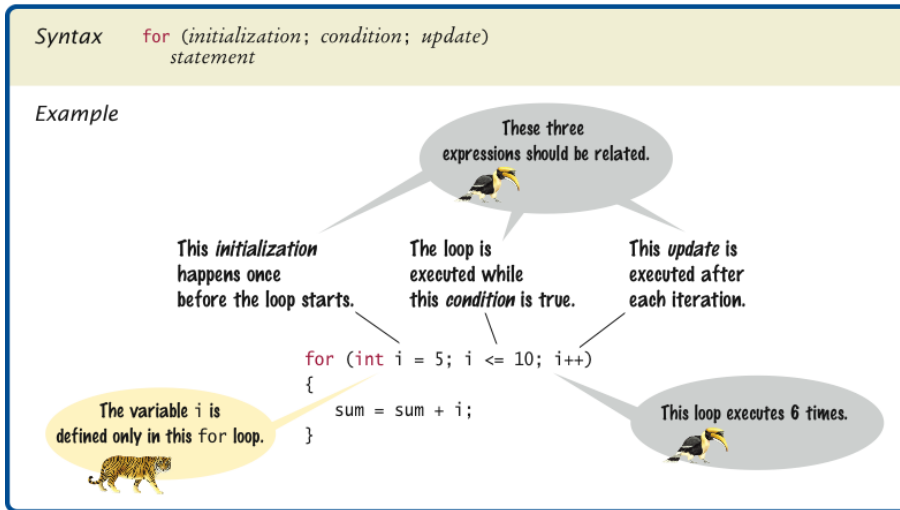
- Example:

```
for (int i = 1; i <= n; i++)
{
    double interest = balance * rate / 100;
    balance = balance + interest;
}
```

- Use a `for` loop when a variable runs from a starting value to an ending value with a constant increment or decrement

Big Java by Cay Horstmann
 Copyright © 2009 by John Wiley & Sons. All rights reserved.

Syntax 6.2 The `for` Statement



Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Execution of a `for` Loop

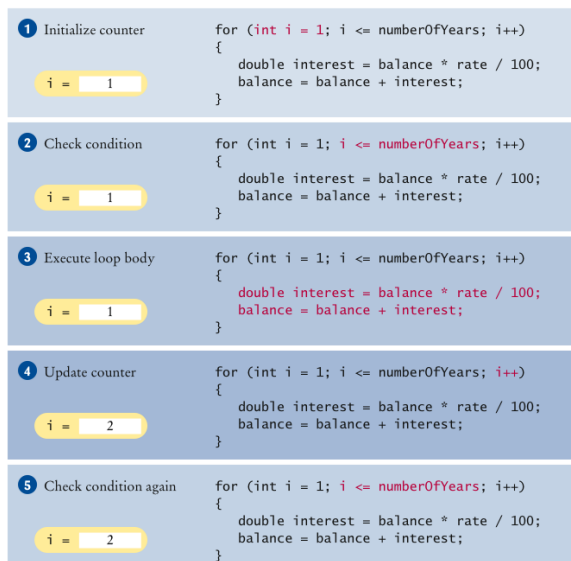
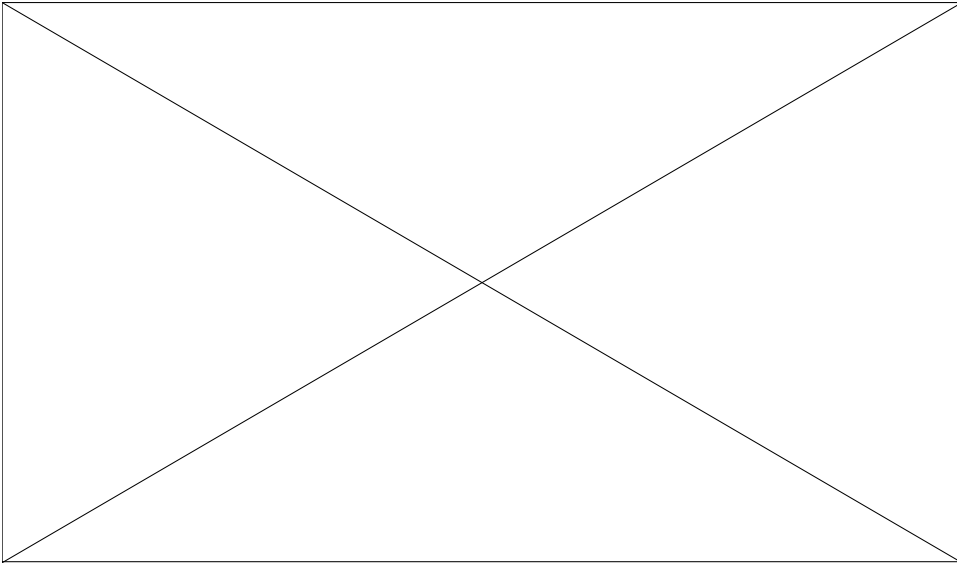


Figure 4 Execution of a `for` Loop

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Animation 6.2: The `for` Loop



Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Self Check 6.4

How many times does the following for loop execute?

```
for (i = 0; i <= 10; i++)  
    System.out.println(i * i);
```

Answer: 11 times.

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

for Loop Examples

Table 2 for Loop Examples

Loop	Values of i	Comment
<code>for (i = 0; i <= 5; i++)</code>	0 1 2 3 4 5	Note that the loop is executed 6 times. (See Quality Tip 6.4 on page 240.)
<code>for (i = 5; i >= 0; i--)</code>	5 4 3 2 1 0	Use <code>i--</code> for decreasing values.
<code>for (i = 0; i < 9; i = i + 2)</code>	0 2 4 6 8	Use <code>i = i + 2</code> for a step size of 2.
<code>for (i = 0; i != 9; i = i + 2)</code>	0 2 4 6 8 10 12 14 ... (infinite loop)	You can use <code><</code> or <code><=</code> instead of <code>!=</code> to avoid this problem.
<code>for (i = 1; i <= 20; i = i * 2)</code>	1 2 4 8 16	You can specify any rule for modifying <code>i</code> , such as doubling it in every step.
<code>for (i = 0; i < str.length(); i++)</code>	0 1 2 ... until the last valid index of the string <code>str</code>	In the loop body, use the expression <code>str.charAt(i)</code> to get the <code>i</code> th character.

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Common Errors: Semicolons

- A missing semicolon:

```
for (years = 1;
    (balance = balance + balance * rate / 100) < targetBalance;
    years++)
    System.out.println(years);
```

- A semicolon that shouldn't be there:

```
sum = 0;
for (i = 1; i <= 10; i++);
    sum = sum + i;
System.out.println(sum);
```

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Loop and a Half

- Sometimes termination condition of a loop can only be evaluated in the middle of the loop
- Then, introduce a boolean variable to control the loop:

```

boolean done = false;
while (!done)
{
    Print prompt
    String input = read input;
    if (end of input indicated)
        done = true;
    else
    {
        Process input
    }
}

```

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Nested Loops

- Create triangle shape:

```

[]
[][]
[][][]
[][][][]

```

- Loop through rows:

```

for (int i = 1; i <= n; i++)
{
    // make triangle row
}

```

- *Make triangle row* is another loop:

```

for (int j = 1; j <= i; j++)
    r = r + "[";
r = r + "\n";

```

- Put loops together → Nested loops

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Nested Loop Examples

Table 3 Nested Loop Examples

Nested Loops	Output	Explanation
<pre>for (i = 1; i <= 3; i++) { for (j = 1; j <= 4; j++) { Print "*" } System.out.println(); }</pre>	<pre>**** **** ****</pre>	Prints 3 rows of 4 asterisks each.
<pre>for (i = 1; i <= 4; i++) { for (j = 1; j <= 3; j++) { Print "*" } System.out.println(); }</pre>	<pre>*** *** *** ***</pre>	Prints 4 rows of 3 asterisks each.

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Nested Loop Examples

Table 3 Nested Loop Examples, continued

Nested Loops	Output	Explanation
<pre>for (i = 1; i <= 4; i++) { for (j = 1; j <= i; j++) { Print "*" } System.out.println(); }</pre>	<pre>* ** *** ****</pre>	Prints 4 rows of lengths 1, 2, 3, and 4.
<pre>for (i = 1; i <= 3; i++) { for (j = 1; j <= 5; j++) { if (j % 2 == 0) { Print "*" } else { Print "-" } } System.out.println(); }</pre>	<pre>-*- -*-* -***-</pre>	Prints asterisks in even columns, dashes in odd columns.
<pre>for (i = 1; i <= 3; i++) { for (j = 1; j <= 5; j++) { if ((i + j) % 2 == 0) { Print "*" } else { Print " " } } System.out.println(); }</pre>	<pre>* * * * * * * *</pre>	Prints a checkerboard pattern.

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Self Check 6.9

How would you modify the nested loops so that you print a square instead of a triangle?

Answer: Change the inner loop to

```
for (int j = 1; j <= width; j++)
```

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.

Self Check 6.10

What is the value of `n` after the following nested loops?

```
int n = 0;
for (int i = 1; i <= 5; i++)
    for (int j = 0; j < i; j++)
        n = n + j;
```

Answer: 20.

Big Java by Cay Horstmann
Copyright © 2009 by John Wiley & Sons. All rights reserved.