

Chapter 4

Loops

Motivations

Suppose that you need to print a string (e.g., "Welcome to Java!") a hundred times. It would be tedious to have to write the following statement a hundred times:

System.out.println("Welcome to Java!");

So, how do you solve this problem?

Liang, Introduction to Java Programming,
Eighth Edition, (c) 2011 Pearson Education, Inc. All
rights reserved. 0132130807

1

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All
rights reserved. 0132130807

2

Opening Problem

Problem:

100
times {

```
System.out.println("Welcome to Java!");
System.out.println("Welcome to Java!");
System.out.println("Welcome to Java!");
System.out.println("Welcome to Java!");
System.out.println("Welcome to Java!");
System.out.println("Welcome to Java!");
...
...
...
System.out.println("Welcome to Java!");
System.out.println("Welcome to Java!");
System.out.println("Welcome to Java!");
```

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All
rights reserved. 0132130807

3

Introducing while Loops

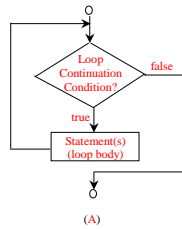
```
int count = 0;
while (count < 100) {
    System.out.println("Welcome to Java");
    count++;
}
```

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All
rights reserved. 0132130807

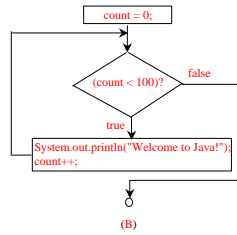
4

while Loop Flow Chart

```
while (loop-continuation-condition) {
    // loop-body;
    Statement(s);
}
```



```
int count = 0;
while (count < 100) {
    System.out.println("Welcome to Java!");
    count++;
}
```



Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

5

Trace while Loop

```
int count = 0;
```

Initialize count

```
while (count < 2) {
    System.out.println("Welcome to Java!");
    count++;
}
```

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

6

Trace while Loop, cont.

```
int count = 0;
```

```
while (count < 2) {
    System.out.println("Welcome to Java!");
    count++;
}
```

(count < 2) is true

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

7

Trace while Loop, cont.

```
int count = 0;
```

```
while (count < 2) {
    System.out.println("Welcome to Java!");
    count++;
}
```

Print Welcome to Java

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

8

Trace while Loop, cont.

```
int count = 0;
```

```
while (count < 2) {
```

```
    System.out.println("Welcome to Java!");
```

```
    count++;
```

```
}
```

Increase count by 1
count is 1 now

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

9

Trace while Loop, cont.

```
int count = 0;
```

```
while (count < 2) {
```

```
    System.out.println("Welcome to Java!");
```

```
    count++;
```

```
}
```

(count < 2) is still true since count
is 1

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

10

Trace while Loop, cont.

```
int count = 0;
```

```
while (count < 2) {
```

```
    System.out.println("Welcome to Java!");
```

```
    count++;
```

```
}
```

Print Welcome to Java

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

11

Trace while Loop, cont.

```
int count = 0;
```

```
while (count < 2) {
```

```
    System.out.println("Welcome to Java!");
```

```
    count++;
```

```
}
```

Increase count by 1
count is 2 now

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

12

Trace while Loop, cont.

```
int count = 0;
while (count < 2) {
    System.out.println("Welcome to Java!");
    count++;
}
```

(count < 2) is false since count is 2 now

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

13

Trace while Loop

```
int count = 0;
while (count < 2) {
    System.out.println("Welcome to Java!");
    count++;
}
```

The loop exits. Execute the next statement after the loop.

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

14

Ending a Loop with a Sentinel Value

- Often the number of times a loop is executed is not predetermined. You may use an input value to signify the end of the loop. Such a value is known as a *sentinel value*.

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

15

Caution

- Don't use floating-point values for equality checking in a loop control. Since floating-point values are approximations for some values, using them could result in imprecise counter values and inaccurate results. Consider the following code for computing $1 + 0.9 + 0.8 + \dots + 0.1$:

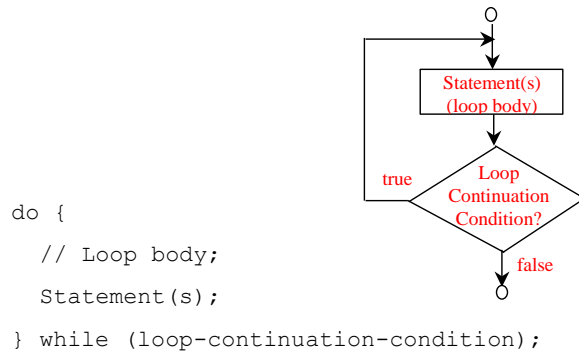
```
double item = 1; double sum = 0;
while (item != 0) { // No guarantee item will be 0
    sum += item;
    item -= 0.1;
}
System.out.println(sum);
```

Variable `item` starts with 1 and is reduced by 0.1 every time the loop body is executed. The loop should terminate when `item` becomes 0. However, there is no guarantee that `item` will be exactly 0, because the floating-point arithmetic is approximated. This loop seems OK on the surface, but it is actually an infinite loop.

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

16

do-while Loop



Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

17

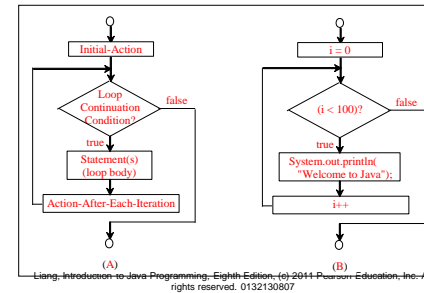
for Loops

```

for (initial-action; loop-
continuation-condition;
action-after-each-iteration) {
    // loop body;
    Statement(s);
}
  
```

```

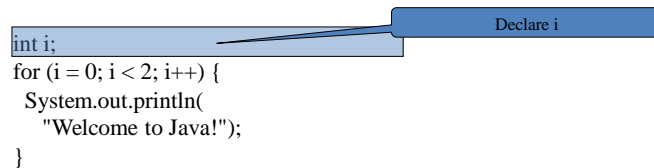
int i;
for (i = 0; i < 100; i++) {
    System.out.println(
        "Welcome to Java!");
}
  
```



Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

18

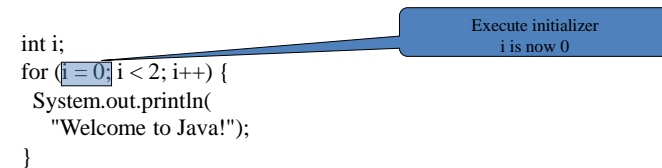
Trace for Loop



Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

19

Trace for Loop, cont.



Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

20

Trace for Loop, cont.

```
int i;
for (i = 0; i < 2; i++) {
    System.out.println( "Welcome to Java!");
}
```

(i < 2) is true
since i is 0

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

21

Trace for Loop, cont.

```
int i;
for (i = 0; i < 2; i++) {
    System.out.println("Welcome to Java!");
}
```

Print Welcome to Java

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

22

Trace for Loop, cont.

```
int i;
for (i = 0; i < 2; i++) {
    System.out.println("Welcome to Java!");
}
```

Execute adjustment statement
i now is 1

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

23

Trace for Loop, cont.

```
int i;
for (i = 0; i < 2; i++) {
    System.out.println("Welcome to Java!");
}
```

(i < 2) is still true
since i is 1

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

24

Trace for Loop, cont.

```
int i;
for (i = 0; i < 2; i++) {
    System.out.println("Welcome to Java!");
}
```

Print Welcome to Java

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

25

Trace for Loop, cont.

```
int i;
for (i = 0; i < 2; i++) {
    System.out.println("Welcome to Java!");
}
```

Execute adjustment statement
i now is 2

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

26

Trace for Loop, cont.

```
int i;
for (i = 0; i < 2; i++) {
    System.out.println("Welcome to Java!");
}
```

(i < 2) is false
since i is 2

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

27

Trace for Loop, cont.

```
int i;
for (i = 0; i < 2; i++) {
    System.out.println("Welcome to Java!");
}
```

Exit the loop. Execute the next
statement after the loop

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

28

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++));
```

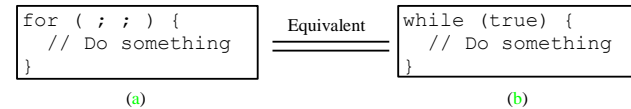
```
for (int i = 0, j = 0; (i + j < 10); i++, j++) {  
    // Do something  
}
```

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

29

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:



Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

30

Caution

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

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

Logic
Error

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

31

Caution, cont.

Similarly, the following loop is also wrong:

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

Logic Error

In the case of the do loop, the following semicolon is needed to end the loop.

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

Correct

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

32

Which Loop to Use?

The three forms of loop statements, while, do-while, and for, are expressively equivalent; that is, you can write a loop in any of these three forms. For example, a while loop in (a) in the following figure can always be converted into the following for loop in (b):

<pre>while (loop-continuation-condition) { // Loop body }</pre>	\equiv Equivalent	<pre>for (; loop-continuation-condition;) // Loop body</pre>
(a)		(b)

A for loop in (a) in the following figure can generally be converted into the following while loop in (b) except in certain special cases (see Review Question 3.19 for one of them):

<pre>for (initial-action; loop-continuation-condition; action-after-each-iteration) { // Loop body; }</pre>	\equiv Equivalent	<pre>initial-action; while (loop-continuation-condition) { // Loop body; action-after-each-iteration; }</pre>
(a)		(b)

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

33

Recommendations

Use the one that is most intuitive and comfortable for you. In general, a for loop may be used if the number of repetitions is known, as, for example, when you need to print a message 100 times. A while loop may be used if the number of repetitions is not known, as in the case of reading the numbers until the input is 0. A do-while loop can be used to replace a while loop if the loop body has to be executed before testing the continuation condition.

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

34

Nested Loops

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

Liang, Introduction to Java Programming, Eighth Edition, (c) 2011 Pearson Education, Inc. All rights reserved. 0132130807

35