

Monday February 4  
Lecture 9

# for-loop: flow chart

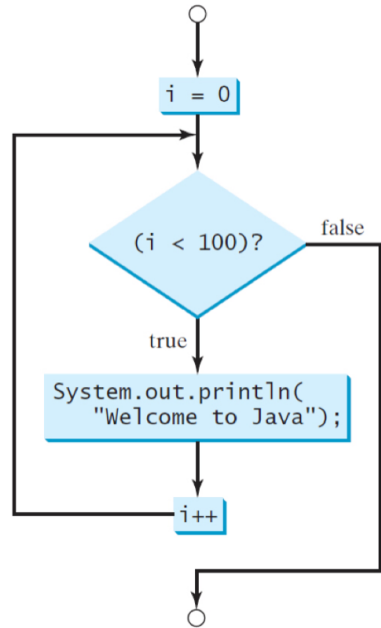
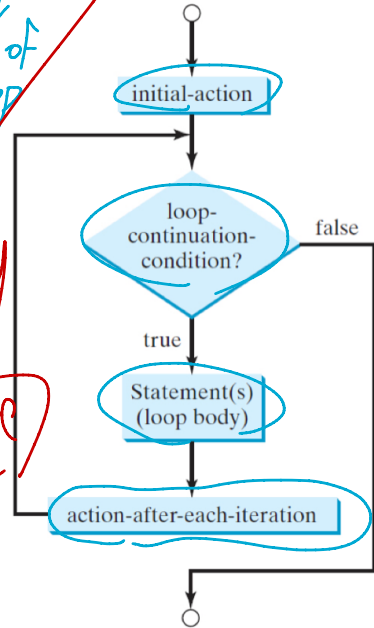
stay condition

update at the end

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

body of loop

eventually should be false



101 times  
↳ 100 times (T)  
↳ 1st (F)

How many times is "i < 100" checked?  
How many times is "println(...)" executed?

# for-loop : Tracing

True → stay in loop  
False → exit from loop

$i < 100$

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

0  
:  
99  
100  
 $100 < 100$

| i   | i < 100   | Enter/Stay Loop? | Iteration | Actions     |
|-----|-----------|------------------|-----------|-------------|
| 0   | 0 < 100   | True             | 1         | print, i ++ |
| 1   | 1 < 100   | True             | 2         | print, i ++ |
| 2   | 2 < 100   | True             | 3         | print, i ++ |
| ... |           |                  |           |             |
| 99  | 99 < 100  | True             | 100       | print, i ++ |
| 100 | 100 < 100 | False            | -         | -           |

How many times is "i < 100" checked?  
How many times is "println(...)" executed?

[0, 99] 100 times  
execute body  
of loop

## for-Loop: Alternative Syntax

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

Annotations:  
- "executed once" points to the initialization part `int i = 0`.  
- "executed at the end of iteration." points to the increment part `i++`.

- The *"initial-action"* is executed *only once*, so it may be moved right before the for loop.
- The *"action-after-each-iteration"* is executed repetitively to *make progress*, so it may be moved to the end of the for loop body.

So the above for-loop may be re-written as:

```
→ int i = 0;  
for ( ; i < 100 ; ) {  
    println ( " - - - " );  
    i++ ;  
}
```

# for-Loop : Exercise (1)

$$\text{Count } (1) = \frac{2 \times i - 1}{1}$$

Compare the behaviour of this program

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

and this program

```
for (int count = 1; count < 201; count += 2) {
    System.out.println("Welcome to Java!");
}
```

$$\text{Count} = 2 \times i - 1$$

$1 = 2 \times 1 - 1$   
 $3 = 2 \times 2 - 1$

Are the outputs same or different?

[1, 3, 5, ..., 199]

| count | count < 100 | Iteration (i) |
|-------|-------------|---------------|
| 0     | T           | 1             |
| 1     | T           | 2             |
| ⋮     |             |               |
| 99    | T           | ?             |
| 100   | 100 < 100 F | 100 + 1       |

| count | count < 201 | Iteration (i) |
|-------|-------------|---------------|
| 1     | 1 < 201 T   | 1             |
| 3     | 3 < 201 T   | 2             |
| 5     | 5 < 201 T   | 3             |
| ⋮     |             |               |
| 199   | 199 < 201 T | ?             |
| 201   | 201 < 201 F | 100           |

$$\text{count} = 2 \times i - 1$$

$199 = 2 \times 100 - 1$

## for-Loop : Exercise (2)

Welcome --- 0  
Welcome --- 1  
Welcome --- 2  
⋮

Compare the behaviour of this program

```
int count = 0;
for (; count < 100; ) {
    System.out.println("Welcome to Java " + count + "!");
    count++; /* count = count + 1; */
}
```

[0, 99] 100 99

and this program

```
int count = 1;
for (; count <= 100; ) {
    System.out.println("Welcome to Java " + count + "!");
    count++; /* count = count + 1; */
}
```

[1, 100] 100

Are the outputs same or different?

# of iterations : 100

Welcome --- 1  
Welcome --- 2  
⋮  
Welcome --- 100

# for-Loop : Exercise (3)

Compare the behaviour of the following three programs:

```
for (int i = 1; i <= 5; i++) {  
    System.out.print(i);  
}
```

Output: 12345

```
int i = 1;  
for ( ; i <= 5; ) {  
    System.out.print(i);  
    i++;  
}
```

Output: 12345

```
int i = 2;  
for ( ; i <= 5; ) {  
    i++;  
    System.out.print(i);  
}
```

Output: 23456

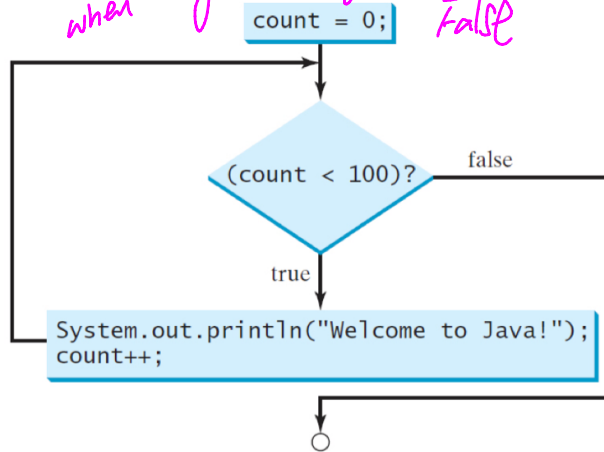
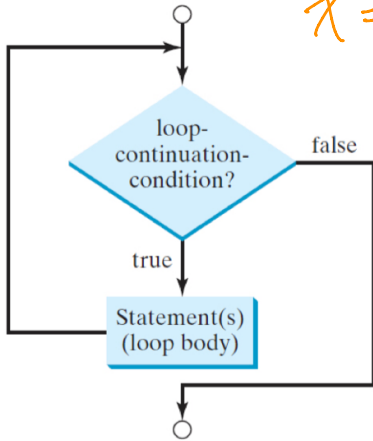
| i | i <= 5   | i++ | print |
|---|----------|-----|-------|
| 1 | 1 <= 5 T | 2   | 2     |
| 2 | 2 <= 5 T | 3   | 3     |
| 3 | 3 <= 5 T | 4   | 4     |
| 4 | 4 <= 5 T | 5   | 5     |
| 5 | 5 <= 5 T | 6   | 6     |
| 6 | 6 <= 5 F |     |       |

# while-loop: flow chart

```
int count = 0;
while (count < 100) {
    System.out.println("Welcome to Java!");
    count++; /* count = count + 1; */
}
```

for ( — ; — ; — )  
while ( — )

$x = y + 1$   
when stay condition evaluates to False



- How many times is "`i < 100`" checked?
- How many times is "`println(...)`" executed?



# while-loop : Tracing

[3, 102]

$$102 - 3 + 1 = \frac{100}{\# \text{ of iterations}}$$

```
int j = 3;
while (j < 103) {
    System.out.println("Welcome to Java!");
    j++; /* j = j + 1; */
}
```

$j - 2 = \text{iteration}$

| j   | j < 103   | Enter/Stay Loop? | iteration | Actions     |
|-----|-----------|------------------|-----------|-------------|
| 3   | 3 < 103   | True             | 1         | print, j ++ |
| 4   | 4 < 103   | True             | 2         | print, j ++ |
| 5   | 5 < 103   | True             | 3         | print, j ++ |
| ... |           |                  |           |             |
| 102 | 102 < 103 | True             | 100       | print, j ++ |
| 103 | 103 < 103 | False            | -         | -           |

How many times is "i < 100" checked?

How many times is "println(...)" executed?

# while-Loop : Exercise (1)

Compare the behaviour of this program

```
int count = 0;
while (count < 100) {
    System.out.println("Welcome to Java!");
    count ++; /* count = count + 1; */
}
```

[0, 99]  
100 times

↓ and this program

```
int count = 1;
while (count <= 100) {
    System.out.println("Welcome to Java!");
    count ++; /* count = count + 1; */
}
```

[1, 100]  
100 times

↓

| count | count < 100 | Iteration (i) |
|-------|-------------|---------------|
|       |             |               |

↓

| count | count <= 100 | Iteration (i) |
|-------|--------------|---------------|
|       |              |               |

## while-Loop : Exercise (2)

Welcome ... 0

Compare the behaviour of this program

```
int count = 0;
→ while (count < 100) {
→ System.out.println("Welcome to Java " + count + "!");
  count ++; /* count = count + 1; */
}
```

and this program



```
int count = 1;
while (count <= 100) {
→ System.out.println("Welcome to Java " + count + "!");
  count ++; /* count = count + 1; */
}
```

Are the outputs same or different?

Welcome ... 1

# Compound Loop: Exercise (1)

count = + 2 X  
count = count + 2  
count += 2

```
System.out.println("Enter a radius value:");  
double radius = input.nextDouble();  
while (radius >= 0) {  
    double area = radius * radius * 3.14;  
    System.out.println("Area is " + area);  
    System.out.println("Enter a radius value:");  
    radius = input.nextDouble();  
    System.out.println("Error: negative radius value.");  
}
```

Test 1:

radius = ~~-3~~

Test 2:

radius = 2

radius = ~~-3~~

Area is -  
Error: neg. radius value

Test 3:

radius = 2

radius = 3