Lecture 2

Recommendations

Use the one that is most intuitive and comfortable for you.

- 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.

Corner Cases 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
}
```

Corner Cases 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)

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);
}
```

Caution, cont.

Similarly, the following loop is also wrong:

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

In the case of the <u>do</u> 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
```

Your Turn! Loop forms

Write three loops, a while loop, do while loop and a for loop that all print all square integers between 1 and 100 on one line:

```
1, 4, 9, 16, 25, … , 100
1, 4, 9, 16, 25, … , 100
1, 4, 9, 16, 25, … , 100
```