Royal University of Bhutan

Unit II

Flow Controls

Tutor: Pema Galey

Learning Outcomes

In this session, you will learn about:

- Java if…else
- Java switch statement
- Java for loop
- Java for-each loop
- Java while loop
- Break statements
- Continue Statements

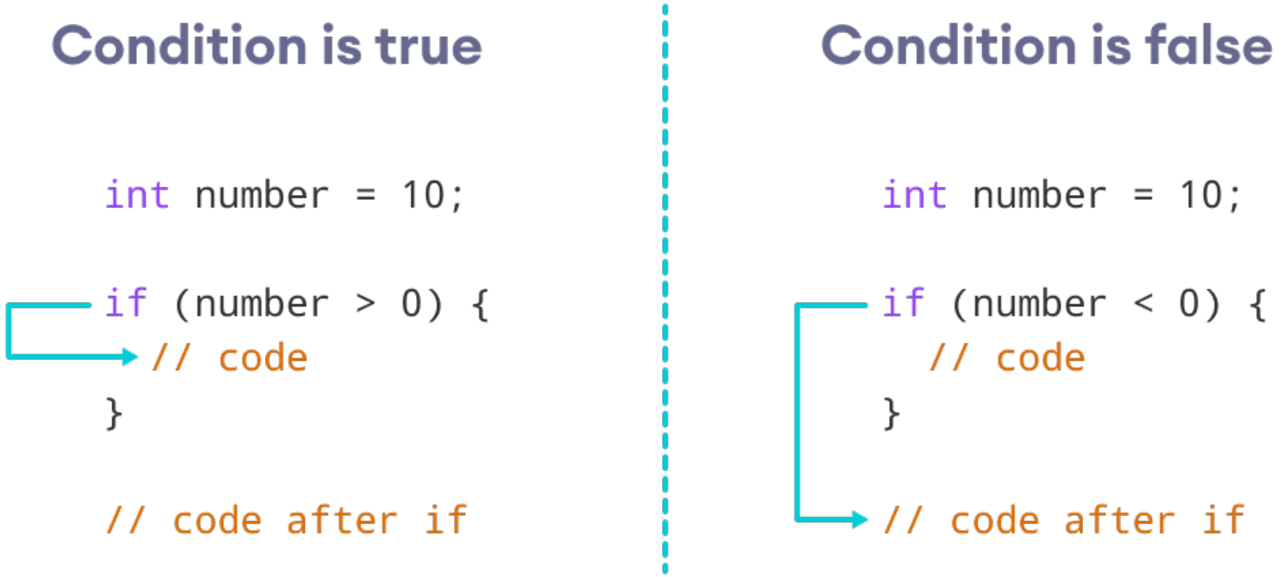
Java if...else Statement

- There are three forms of if…else statements in Java:
  1. if statement
  2. if...else statement
  3. if...else if...else statement
  4. Nested if...else statement

1. Java if (if-then) Statement

- The syntax of a **if-then** statement:

```
if (condition) {
        // statements
}
```



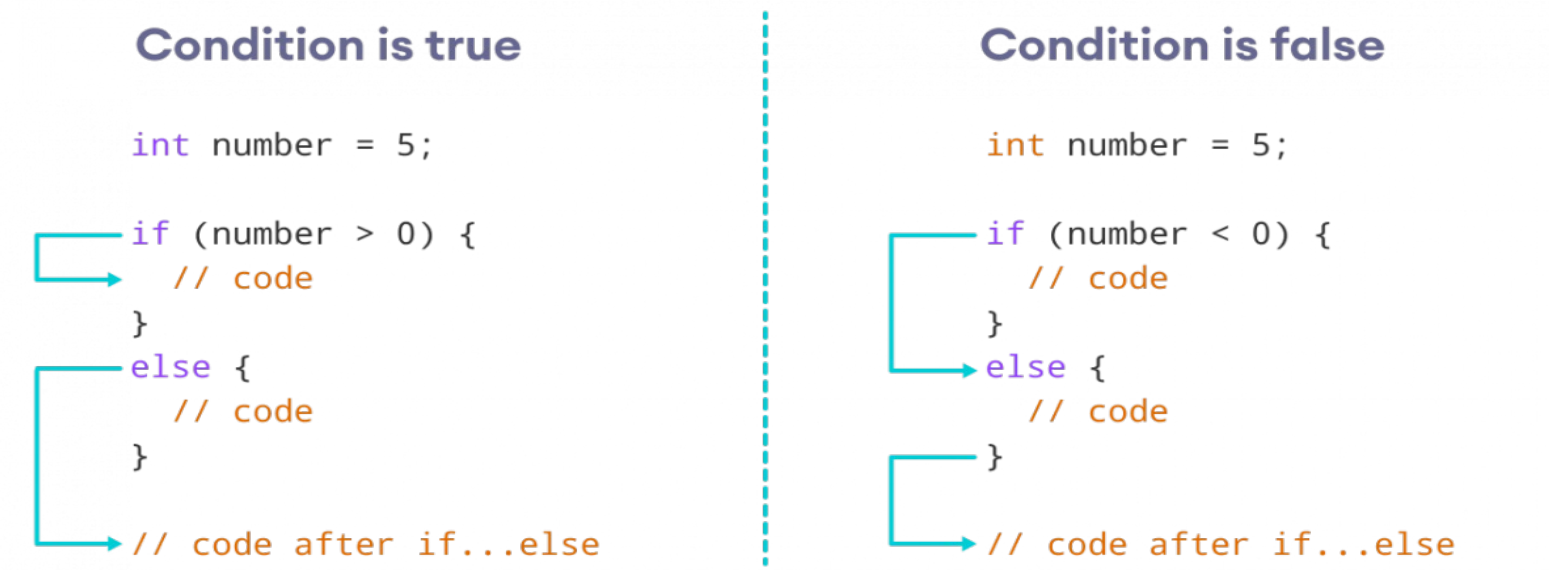
```
int number = 10;

if (number > 0) {
  // code
}

// code after if
```

```
int number = 10;

if (number < 0) {
  // code
}

// code after if
```

2. Java if...else (if-then-else) Statement

- The syntax of the **if...else** statement is:

```
if (condition) {
        // codes in if block
} else { // codes in else block }
```



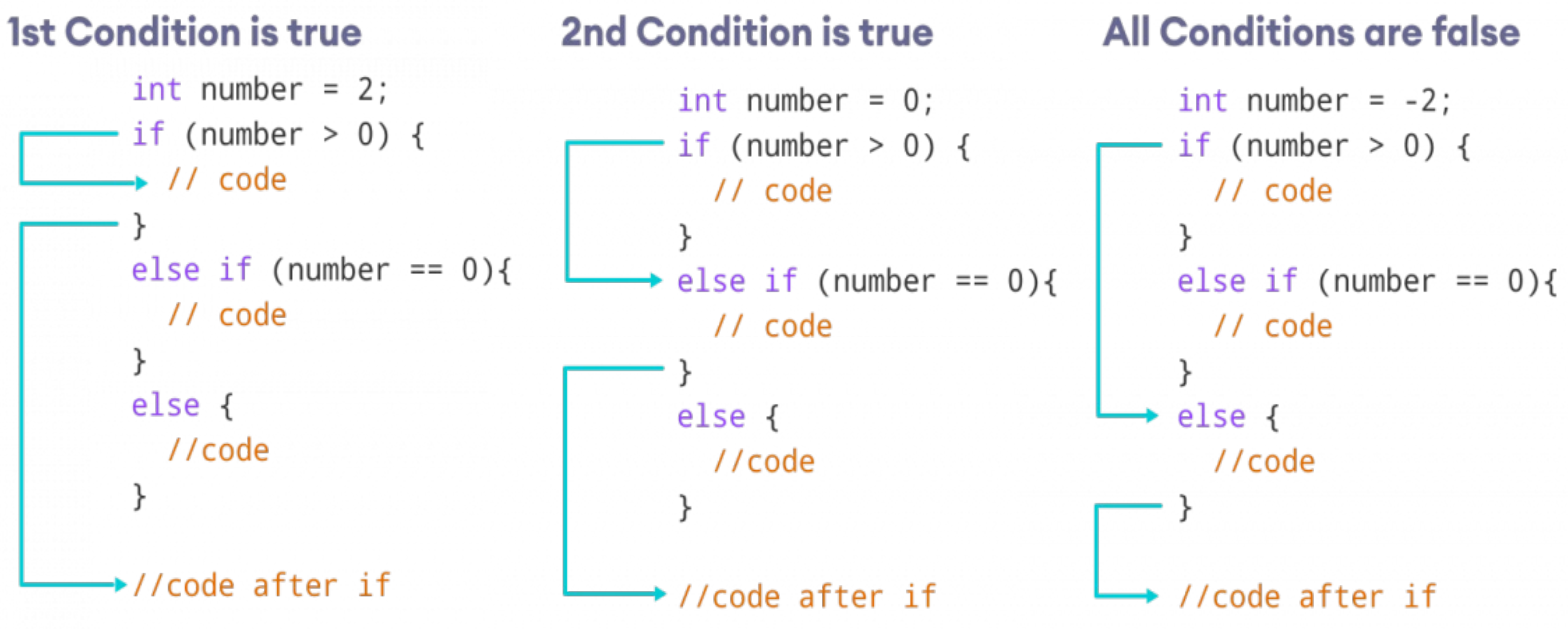
Condition is true

```
int number = 5;

if (number > 0) {
  // code
}
else {
  // code
}

// code after if...else
```

Condition is false

```
int number = 5;

if (number < 0) {
  // code
}
else {
  // code
}

// code after if...else
```

3. Java if...else...if Statement



1st Condition is true

```
int number = 2;
if (number > 0) {
  // code
}
else if (number == 0){
  // code
}
else {
  //code
}

//code after if
```

2nd Condition is true

```
int number = 0;
if (number > 0) {
  // code
}
else if (number == 0){
  // code
}
else {
  //code
}

//code after if
```

All Conditions are false

```
int number = -2;
if (number > 0) {
  // code
}
else if (number == 0){
  // code
}
else {
  //code
}

//code after if
```

4. Java Nested if..else Statement

- In Java, it is also possible to use if..else statements inside an if...else statement. It's called the nested if...else statement.
- Example:

```
// declaring double type variables
    double n1 = -1.0, n2 = 4.5, n3 = -5.3, largest;
    // checks if n1 is greater than or equal to n2
    if (n1 >= n2) {
         // if...else statement inside the if block
         // checks if n1 is greater than or equal to n3
         if (n1 >= n3) {
                  largest = n1;
         } else {
                  largest = n3;
         }
     } else{….. }
```

Java switch Statement

- The switch statement allows us to execute a block of code among many alternatives.
- The syntax of the switch statement in Java is:

```
switch (expression) {
        case value1: // code to be executed if
                // expression is equal to value1
                break;
        case value2: // code to be executed if
                // expression is equal to value2
                break;

        ... ...
        default:
        // default statements
}
```

Example: Switch

1. To find the case for gaming ranks as follows:
   - 1: Winner
   - 2: First Runners up
   - 3: Second Runners up
   - Default: Participants

2. Basic calculator for basic arithematics

Loop

- In computer programming, loops are used to repeat a block of code.
- In Java, there are three types of loops.
  1. for loop
  2. while loop
  3. do...while loop
  4. for each loop

Java for Loop

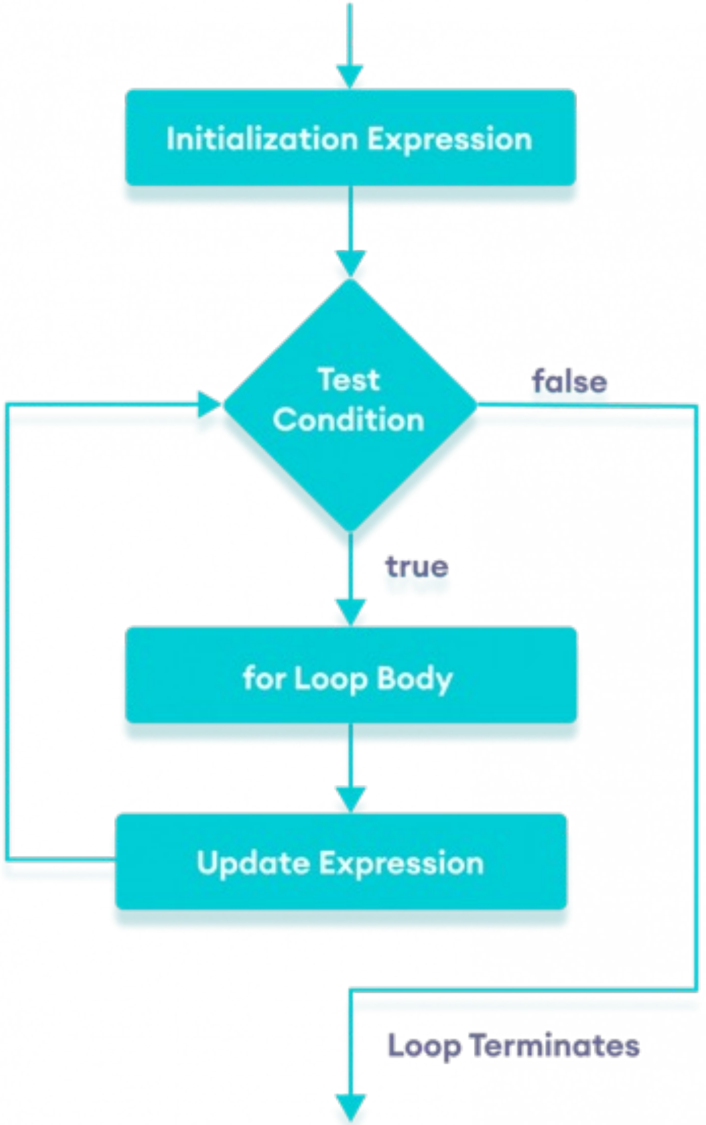
- Java for loop is used to run a block of code for a certain number of times. The syntax of for loop is:

```
for (initialExpression; testExpression; updateExpression) {
        // body of the loop
}
```

Java for Loop

- The **initialExpression** initializes and/or declares variables and executes only once.
- The **condition** is evaluated. If the **condition** is true, the body of the for loop is executed.
- The **updateExpression** updates the value of **initialExpression**.
- The **condition** is evaluated again. The process continues until the **condition** is false.

Java for Loop



Java For loop

- Example:

```
// Print natural numbers from 1 to 5
class Main {
        public static void main(String[] args) {
                // generating natural numbers
                for (int i=1; i<=5; i++) {
                        System.out.println(i);
                }
        }
}
```

Java while Loop

- Java while loop is used to run a specific code until a certain condition is met.
- The syntax of the while loop is:

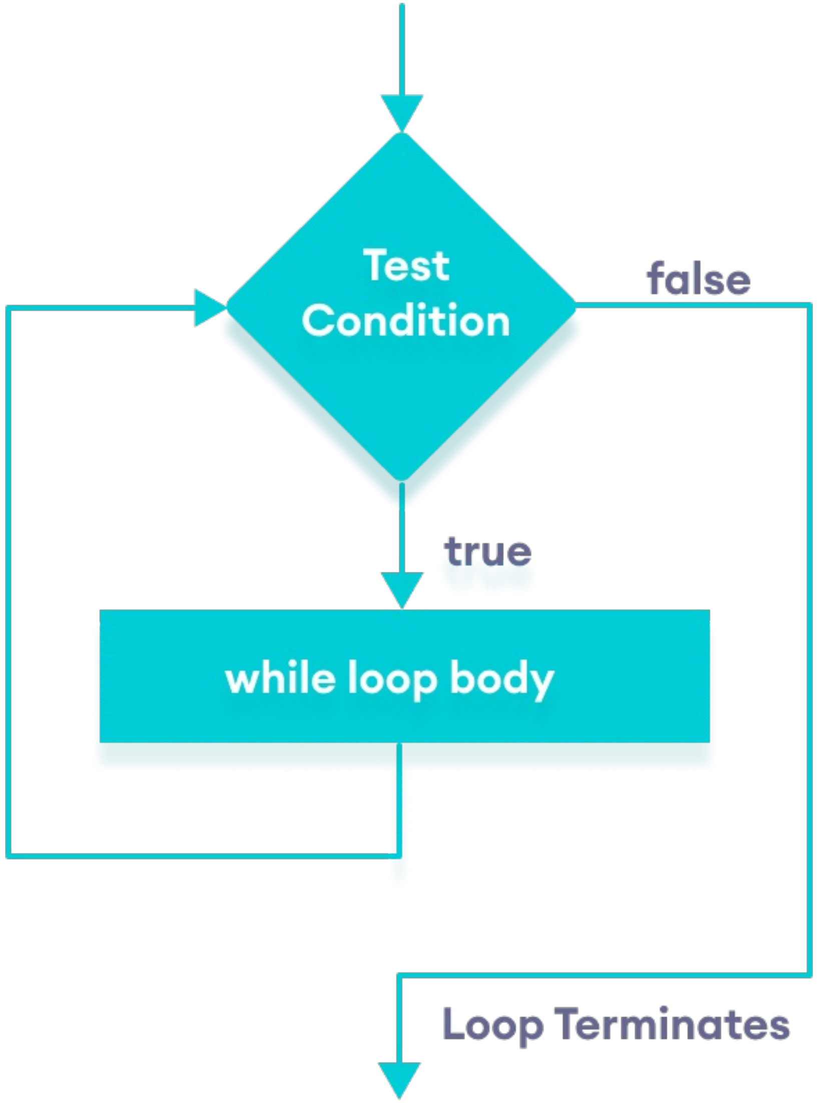
```
while (testExpression) {
        // body of loop
}
```

- Example: To display numbers from 1 to 5

Java while Loop

- Explanation:
  1. A while loop evaluates the **textExpression** inside the parenthesis ().
  2. If the **textExpression** evaluates to true, the code inside the while loop is executed.
  3. The **textExpression** is evaluated again.
  4. This process continues until the **textExpression** is false.
  5. When the **textExpression** evaluates to false, the loop stops.

Java while Loop



Java do-while Loop

- The **do...while** loop is similar to while loop.
- However, the body of **do...while** loop is executed once before the test expression is checked.
- Syntax:

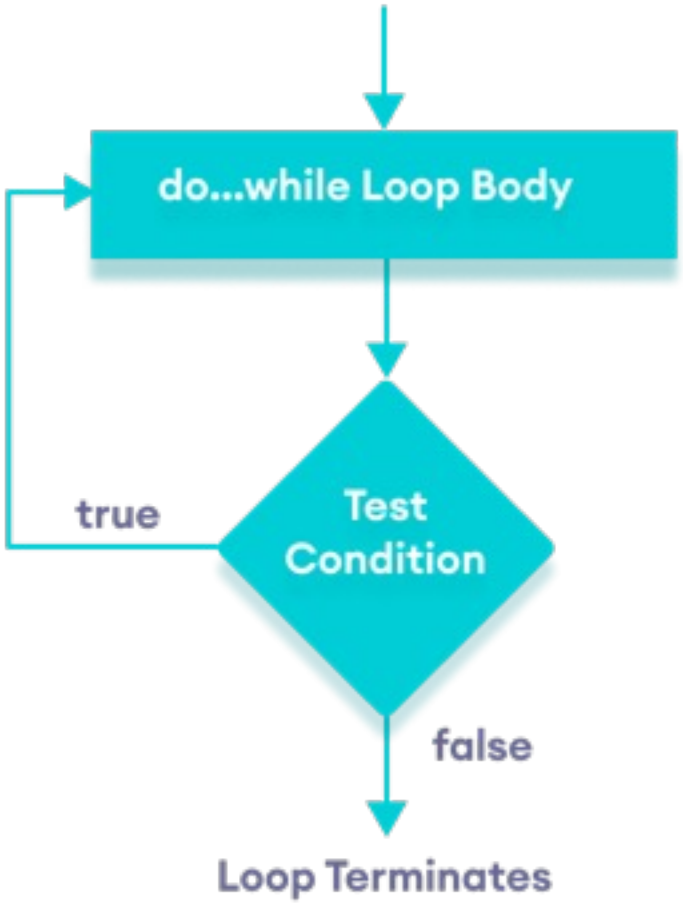
```
 do {
        // body of loop
} while(textExpression);
```

- Example: Display numbers from 1 to 5

Java do-while Loop

- Explanation:
    1. The body of the loop is executed at first. Then the **textExpression** is evaluated.
    2. If the **textExpression** evaluates to true, the body of the loop inside the do statement is executed again.
    3. The **textExpression** is evaluated once again.
    4. If the **textExpression** evaluates to true, the body of the loop inside the do statement is executed again.
    5. This process continues until the **textExpression** evaluates to false. Then the loop stops.

Java do-while Loop



for vs while Loops

- The **for** loop is used when the number of iterations is known. For example,

```
for (let i = 1; i <=5; ++i) {
        // body of loop
}
```

- And **while** and **do...while** loops are generally used when the number of iterations is unknown. For example,

```
while (condition) {
        // body of loop
}
```

Java for-each Loop

- The Java for loop has an alternative syntax that makes it easy to iterate through arrays and collections (ArrayLIst).
- It is also known as the enhanced for loop.
- The syntax of the Java for-each loop is:

```
for(dataType item : array) {
        ...
}
```

Java for-each Loop

- Example :

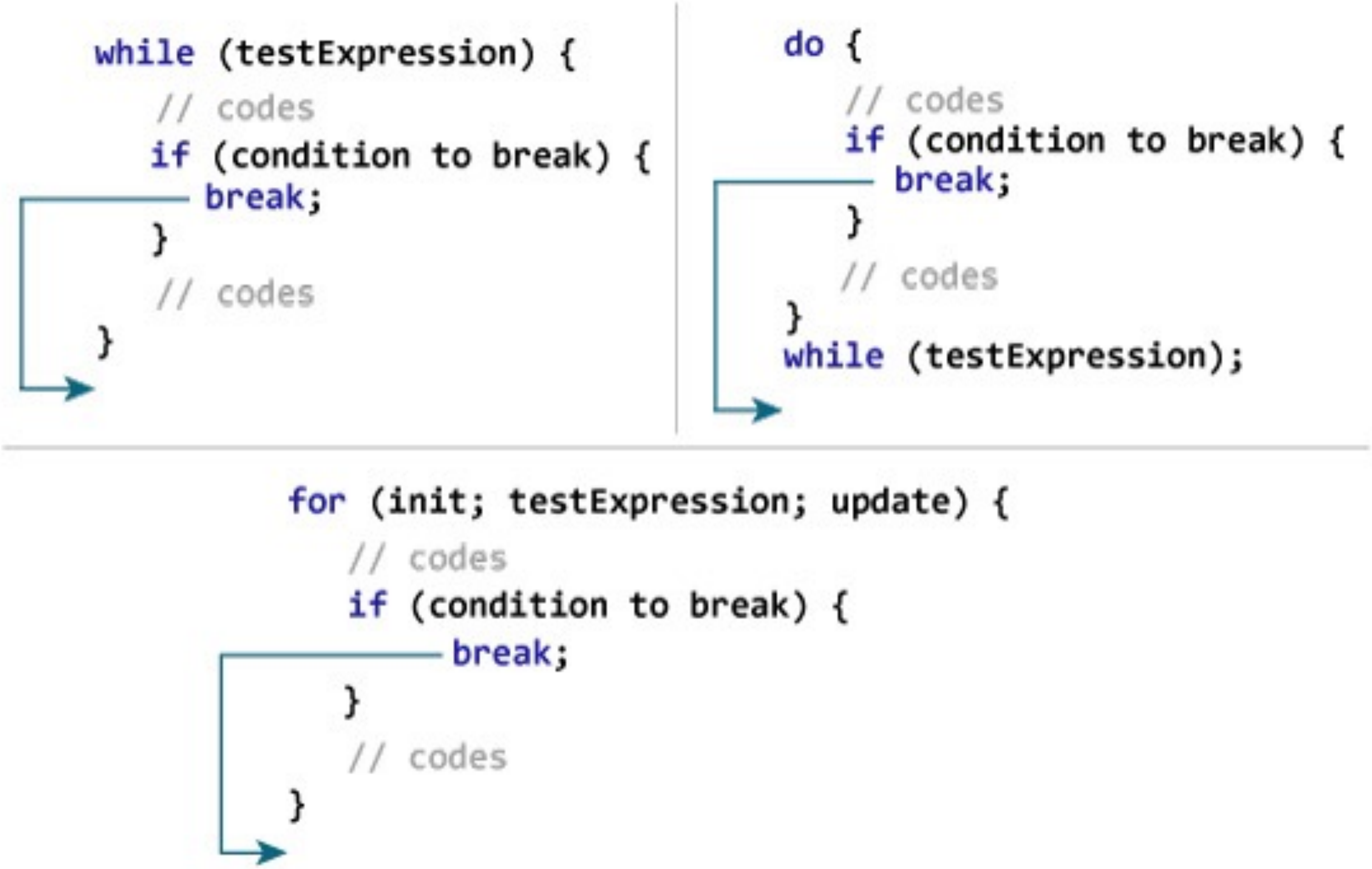
```
// print array elements
class Main {
        public static void main(String[] args) {
                // create an array
                int[] numbers = {3, 7, 5, -5};
                // iterating through the array
                for (int num: numbers) {
                System.out.println(num);
                }
        }
}
```

Break Statement

- The break statement in Java terminates the loop immediately, and the control of the program moves to the next statement following the loop.
- It is almost always used with decision-making statements (if...else Statement).
- Syntax:

```
break;
```

Break Statement



```
while (testExpression) {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
```

```
do {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
while (testExpression);
```

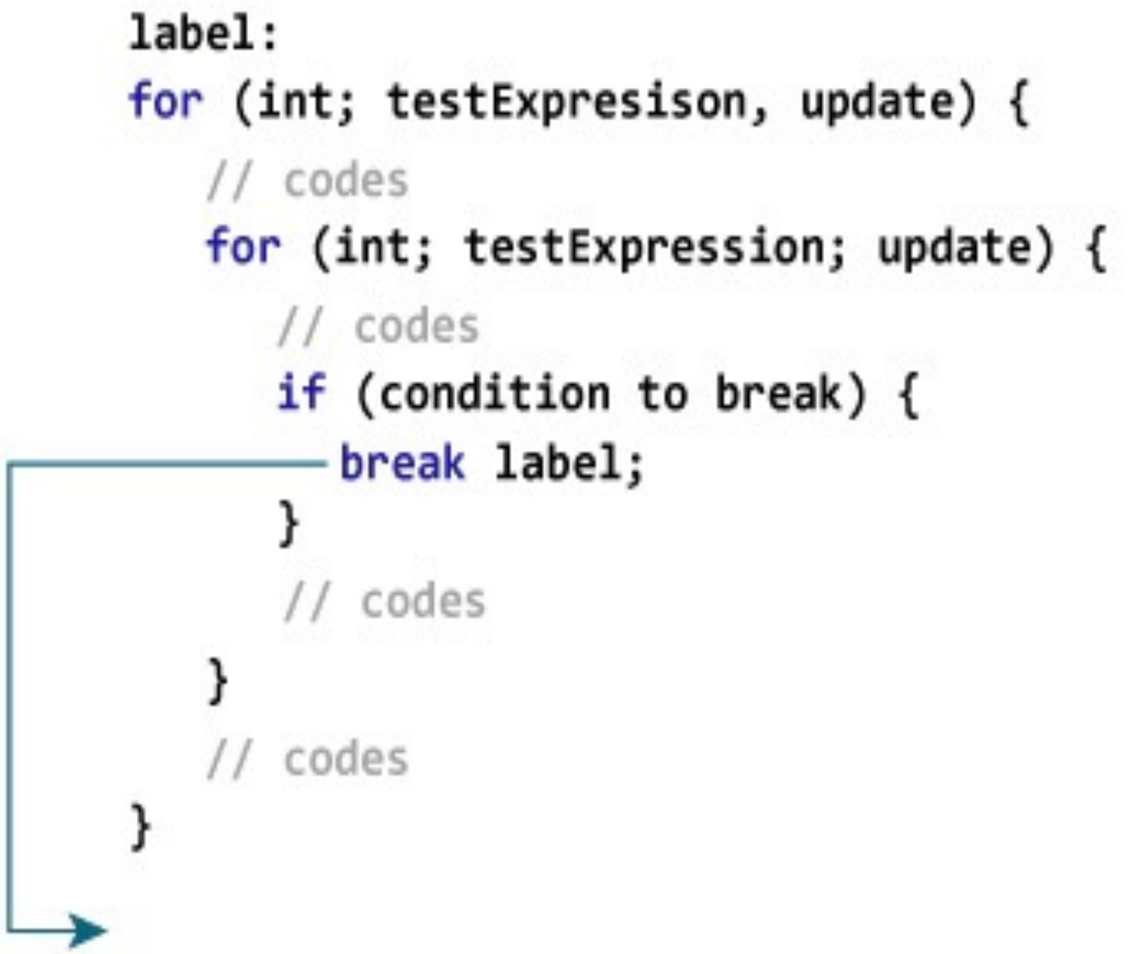
```
for (init; testExpression; update) {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
```

Labeled break Statement

- It terminates the innermost loop and switch statement. However, there is another form of break statement in Java known as the labeled break.
- We can use the labeled break statement to terminate the outermost loop as well.
- Syntax:

```
break label;
```

Labeled break Statement



```
label:
for (int; testExpresison, update) {
    // codes
    for (int; testExpression; update) {
        // codes
        if (condition to break) {
            break label;
        }
        // codes
    }
    // codes
}
```

Labeled break Statement

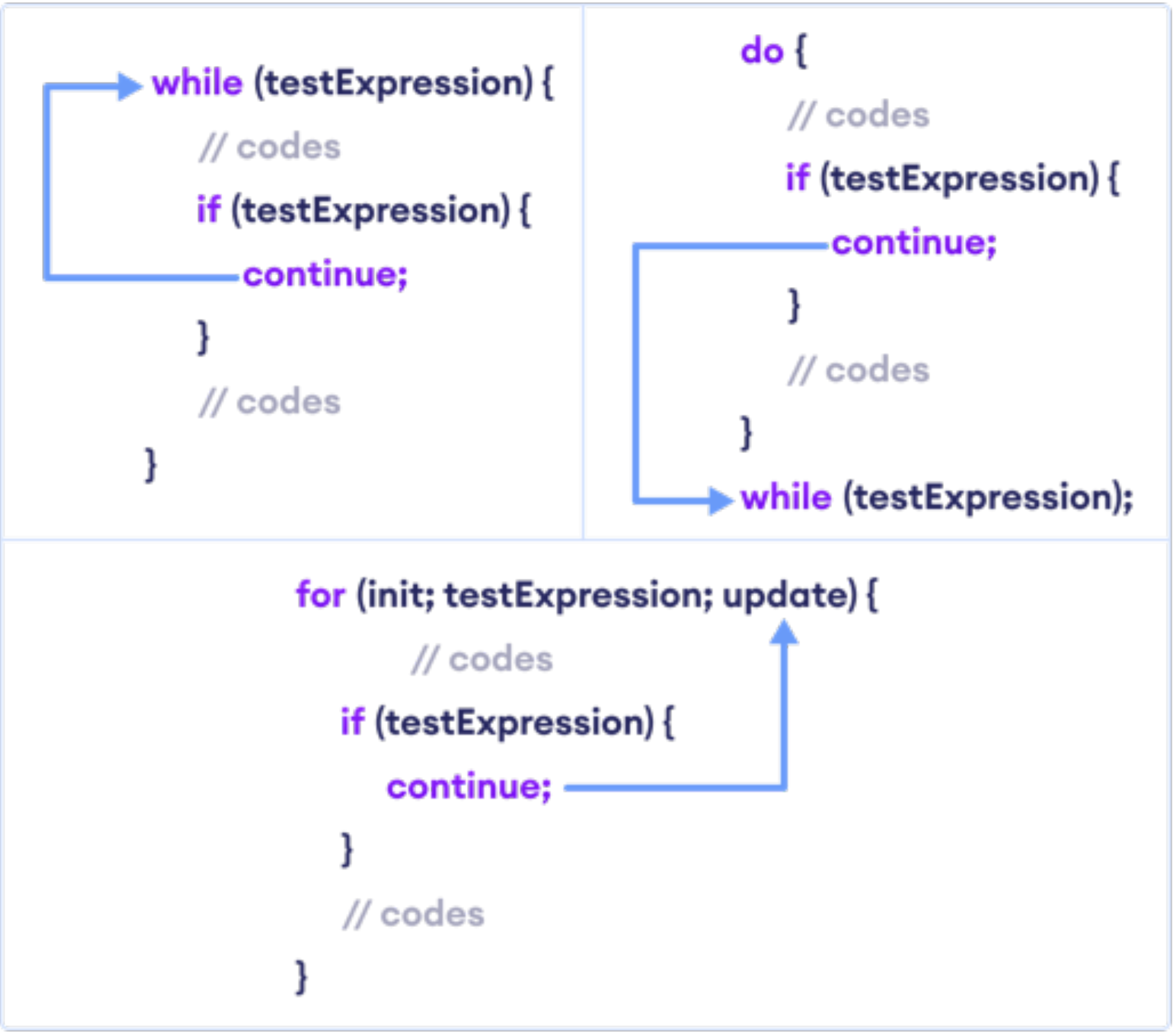
```
class LabeledBreak {
    public static void main(String[] args) {
        // the for loop is labeled as first
        first:
        for( int i = 1; i < 5; i++) {
          // the for loop is labeled as second
          second:
          for(int j = 1; j < 3; j ++ ) {
                System.out.println("i = " + i + "; j = " +j);
                // the break statement breaks the first for loop
                if ( i == 2)
                        break first;
          }
} } }
```

Continue Statement

- The continue statement skips the current iteration of a loop (for, while, do...while, etc).
- After the continue statement, the program moves to the end of the loop. And, test expression is evaluated (update statement is evaluated in case of the for loop).
- Syntax:

```
continue;
```

Continue Statement



```
while (testExpression) {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
```

```
do {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
while (testExpression);
```

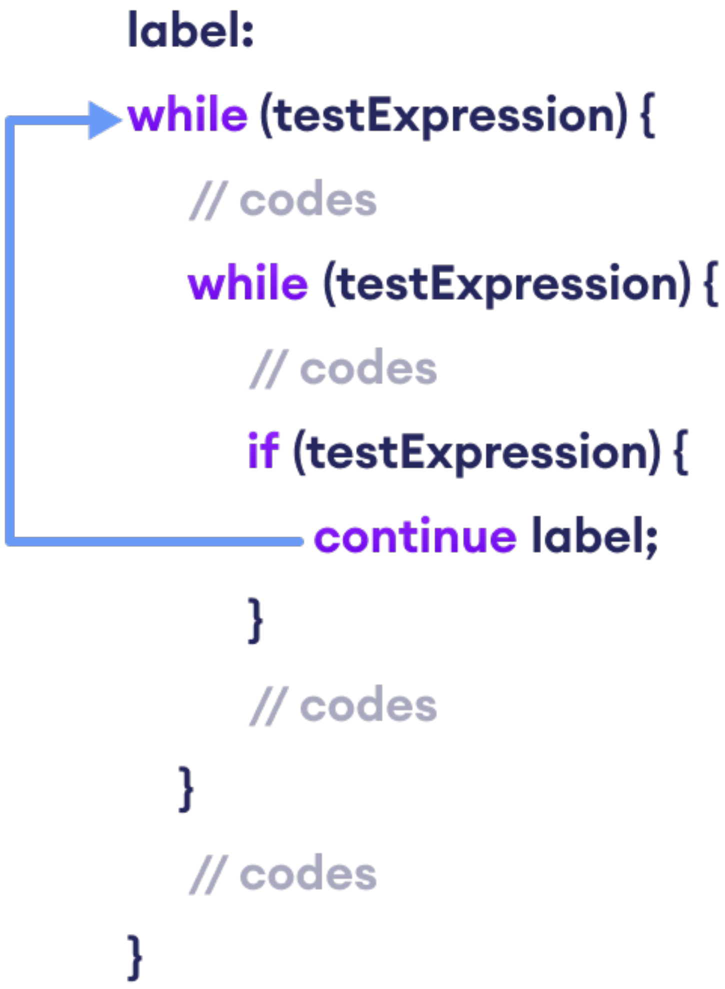
```
for (init; testExpression; update) {
    // codes
    if (testExpression) {
        continue;
    }
    // codes
}
```

Labeled continue Statement

- Till now, we have used the unlabeled continue statement. However, there is another form of continue statement in Java known as labeled continue.
- It includes the label of the loop along with the continue keyword.
- Syntax:

```
continue label;
```

Labeled continue Statement



```
label:
while (testExpression) {
   // codes
   while (testExpression) {
      // codes
      if (testExpression) {
         continue label;
      }
      // codes
   }
   // codes
}
```

```
class Main {
   public static void main(String[] args) {
      // outer loop is labeled as first
      first:
      for (int i = 1; i < 6; ++i) {
          // inner loop
          for (int j = 1; j < 5; ++j) {
             if (i == 3 || j == 2)
               // skips the current
               //iteration of outer loop
              continue first;

             System.out.println("i = "+i+"; j = "+j);
          }
      }
    }
}
```

Thank you!