Looping 2

Basic Programming Teaching Team 2022

Objectives

After studying this material, students should be able to:

- Understand the concept of nested loops in programming
- Get to know nested loop syntax
- Implement nested loops to solve the problem

Preface

- In the basic concept of looping, looping logic is used to perform several **process / program** statements **repeatedly**, with a **certain pattern**.
- Process / statement will continue to be **executed** repeatedly, as long as **the loop condition is true**. Otherwise, the loop will **stop** and the process / statement will not be executed again when **the loop condition is false**.
- **A looping condition** is needed to determine whether a loop will continue or should stop.

Definition

- A nested loop is a loop structure that lies inside another loop.
- In a nested loop, there is an **outer loop** that is in the outermost position and an **inner loop** that is in its inner position.
- A nested loop can consist of more than 2 levels.

Pseudocode

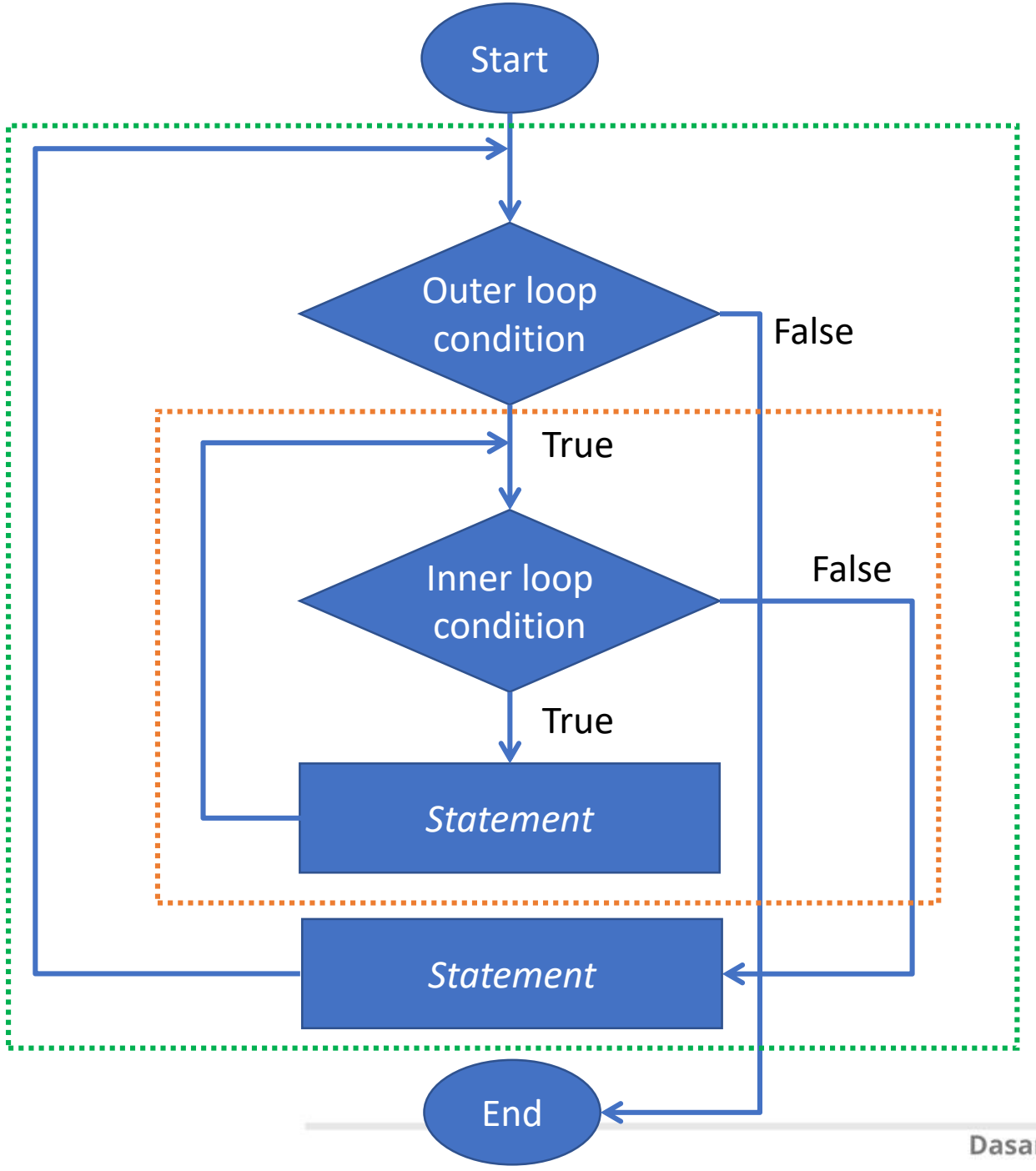
- In general, a nested loop should look like this:

```
loop-level-1 {
    loop-level-2 {
        .....
        loop-level-n {
            // statement
        }
        .....
    }
}
```

Flowchart

Outer loop

Inner loop



Nested Loop Syntax: FOR

```
1  for (int i = 0; i < n; i++){ //loop level 1
2      for (int j = 0; j < n; j++){ //loop level 2
3          for (int k = 0; k < n; k++){ //loop level 3
4              //statement
5          }
6      }
7      for (int l = 0; l < n; l++){ //loop level 2
8          //statement
9      }
10 }
```

Outer loop

Inner loop

Nested Loop Syntax: WHILE

Outer loop

```
1   int i = 0;
2   // Loop checking. As long as condition (i < n) is true, the loop continues
3   while (i < n) { //loop level 1
4       int j = 0;
5
6       // Loop checking. As long as condition (j < n) is true, the loop continues
7       while (j < n) { //loop level 2
8           //statement
9           j++;
10      }
11      i++;
12  }
```

Inner loop

Nested Loop Syntax: DO-WHILE

```
1   int i = 0;
2   do { //loop level 1
3       int j = 0;
4
5       do { //loop level 2
6           //statement
7           j++;
8       }
9       // Loop checking. As long as condition (j < n) is true, the loop continues
10      while (j < n);
11      i++;
12  }
13  // Loop checking. As long as condition (i < n) is true, the loop continues
14  while (i < n);
```

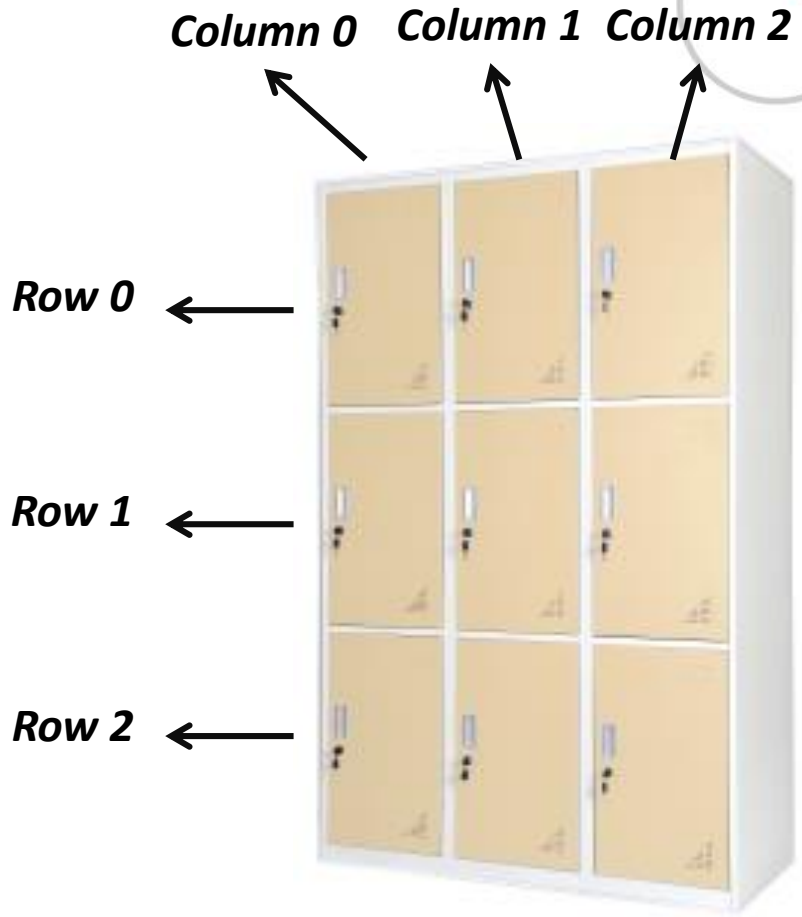
Outer loop

Inner loop

Locker Logic

- A nested loop with 2 levels is like a locker.
- Where the outer loop is identified as a row pointer and the inner loop is identified as a column pointer.

```
for (int row = 0; row < 3; row++){
    for (int column = 0; column < 3; column++){
        //statement
    }
}
```



Locker Logic

```
for (int row = 0; row < 3; row++){
    for (int column = 0; column < 3; column++){
        System.out.print("Row-" + row + " & Column-" + column);
    }
    System.out.println();
}
```

Nested Loop Combinations

- Nested loops not only consist of one type of nested loop, but it can be a combination of nested loops.

for and do-while

```
for(int i = 0; i < 10; i++){
    int j = 0;
    do {
        // statement
        j++;
    } while(j < 10);
}
```

while and do-while

```
int i = 0;
while(i < 10) {
    int j = 0;
    do {
        // statement
        j++;
    } while(j < 10);

    i++;
}
```

while and for

```
int i = 0;
while(i < 10) {
    for(int j = 0; j < 10; j++) {
        // statement
    }
    i++;
}
```

do-while and for

```
int i = 0;
do {
    for(int j = 0; j < 10; j++) {
        // statement
    }
    i++;
} while(i < 10);
```

Case Study

Case Study 1 - Square Star

- How to display * as shown in the image using a nested loop?
- By using nested for, while, and do-while loops

Case Study 1: Answer Logic

- The program to be created must have an **outer loop** and an **inner loop**.
- The **outer loop** is used to count or repeat the number of **rows**, which consists of 4 lines (i = 0; i < 4; i++). Every time the inner loop has finished executing, **a new line will be created**.
- The **inner loop** is used to print the * symbol on the screen, the number of symbols displayed per line will **correspond to the value in variable j**, which consists of 4 symbols * (j = 0; j < 4; j++).

Case Study 1: FOR

```
1  public class StudiKasus1 {
2      public static void main(String args[]) {
3          for (int i = 0; i < 4; i++) {
4              for (int j = 0; j < 4; j++) {
5                  System.out.print("*");
6              }
7              System.out.println();
8          }
9      }
10 }
```

Inner loop produces a row of stars (with 4 stars in each row)

Outer loop repeats the output of the inner loop

Case Study 1: WHILE

```
1  public class StudiKasus1 {
2      public static void main(String args[]) {
3          int i = 0;
4          while (i < 4) {
5
6              int j = 0;
7              while (j < 4) {
8                  System.out.print("*");
9                  j++;
10             }
11             System.out.println();
12             i++;
13         }
14     }
15 }
```

Inner loop produces a row of stars (with 4 stars in each row)

Outer loop repeats the output of the inner loop

Case Study 1: DO-WHILE

```
1  public class StudiKasus1 {
2      public static void main(String args[]) {
3          int i = 0;
4          do {
5
6              int j = 0;
7              do {
8                  System.out.print("*");
9                  j++;
10             }while (j < 4);
11
12             System.out.println();
13             i++;
14         } while (i < 4);
15     }
16 }
```

Inner loop produces a row of stars (with 4 stars in each row)

Outer loop repeats the output of the inner loop

Case Study 2 - Triangle Star

- How to display * as shown in the image using a nested loop?
- The number of stars in a row depends on the row number (10 lines total)

Case Study 2: Answer Logic

- The program to be created must have an **outer loop** and an **inner loop**.
- The **outer loop** is used to count or repeat the number of **rows**, which consists of 10 lines (i = 0; i < 10; i++). Every time the inner loop has finished executing, **a new line will be created**.
- The **inner loop** is used to print the * symbol on the screen, the number of symbols displayed per line will **correspond (equal to) to the row number in i** (j = 0; j <= i; j++).

Case Study 2: FOR

```
1  public class StudiKasus2 {
2      public static void main(String args[]) {
3          for(int i = 0; i < 10; i++) {
4              for(int j = 0; j <= i; j++){
5                  System.out.print("*");
6              }
7              System.out.println();
8          }
9      }
10 }
```

The inner loop generates a row of stars with the number corresponding to the row number in i

Outer loop repeats the output of the inner loop

Case Study 2: WHILE

```
1  public class StudiKasus2 {
2      public static void main(String args[]) {
3          int i = 0;
4          while(i < 10) {
5
6              int j = 0;
7              while(j <= i){
8                  System.out.print("*");
9                  j++;
10             }
11             System.out.println();
12             i++;
13         }
14     }
15 }
```

The inner loop generates a row of stars with the number corresponding to the row number in i

Outer loop repeats the output of the inner loop

Case Study 2: DO-WHILE

```
1  public class StudiKasus2 {
2      public static void main(String args[]) {
3          int i = 0;
4          do {
5              int j = 0;
6              do {
7                  System.out.print("*");
8                  j++;
9              } while(j <= i);
10
11             System.out.println();
12             i++;
13         } while(i < 10);
14     }
15 }
```

The inner loop generates a row of stars with the number corresponding to the row number in i

Outer loop repeats the output of the inner loop

Assignment

Create flowcharts from Case Study 1 and Case Study 2 for all types of loops (for, while, and do-while), but according to the following conditions:

- Students with odd numbers: Case study 1
- Students with even numbers: Case study 2