Lecture 3

switch - case

Loops

While, for, do-while

Keywords ‘break’ and ‘continue’

IT TALENTS
Training Camp

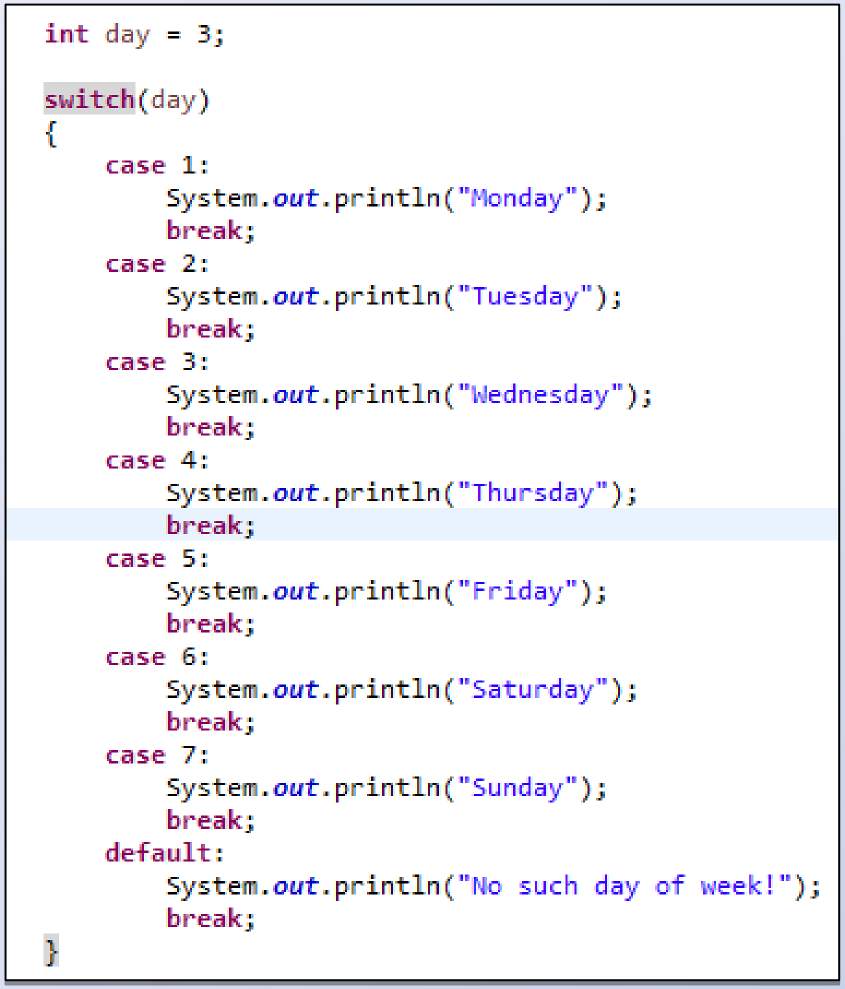
Problems with if-s

- What if we need to compare if a variable holds one of 20 possible values ? 20 If-s ?
- What if we need to execute the same statement when the variable is equal to three different values among those 20 ? Same code executed on several lines ?

IT TALENTS
Training Camp

The ‘switch – case’ statement

- Selects for execution a statement from a list, depending on the value of the switch expression



```
int day = 3;

switch(day)
{
    case 1:
        System.out.println("Monday");
        break;
    case 2:
        System.out.println("Tuesday");
        break;
    case 3:
        System.out.println("Wednesday");
        break;
    case 4:
        System.out.println("Thursday");
        break;
    case 5:
        System.out.println("Friday");
        break;
    case 6:
        System.out.println("Saturday");
        break;
    case 7:
        System.out.println("Sunday");
        break;
    default:
        System.out.println("No such day of week!");
        break;
}
```

How ‘switch – case’ works ?

- The expression is evaluated
- When one of the constants specified in a case label is equal to the expression
    - The statement that corresponds to that case is executed
- If no case is equal to the expression
    - If there is a default case, it is executed
    - Otherwise the control is transferred to the end of the switch statement

IT TALENTS
Training Camp

‘switch – case’ good practices

1) Supported variable types are String, enum and int
2) Only constants supported in cases comparisons
3) break is always a good idea to be used
4) default is always a good idea to be used
5) Multiple cases can execute a single statement
6) Always handle the most probable cases first

IT TALENTS
Training Camp

Problem

Print all the numbers

- From 1 to 5
- From 1 to 1000
- From 1 to n
- From n to m

What is a loop?

- A loop is a structure that allows a sequence of statements to be executed more times in a row
- Loops have a boolean condition and a block of code for execution. While the condition is true, the block is being executed.
- A loop that never ends is called an infinite loop

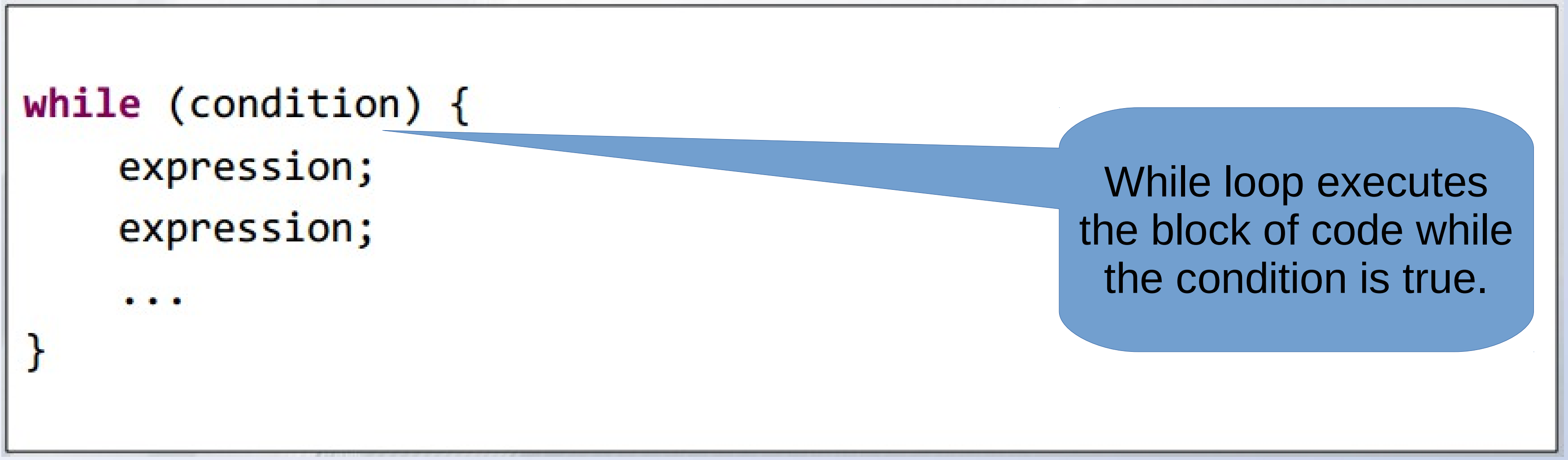
Why we use loops ?

- With loops we can execute similar statements many times
- We gain benefits from the **code reusage**
- Our code becomes much, much simpler

IT TALENTS
Training Camp

While loop

- The while loop is the simplest type of loop in Java.
- However that is not to say that it's not powerful.
- The basic syntax of the while loop is :



```
while (condition) {
     expression;
     expression;
     ...
}
```

While loop executes the block of code while the condition is true.

While loop

- While the condition is true, the block is being executed.

Counter initialization

Boolean condition.
If i > 100, the next block will be skipped

```
int i = 1;

while (i <= 1000) {
    System.out.println(i);
    i++;
}
```

Block of code
repeatable execution

IT TALENTS
Training Camp

Do-While loop

- The do-while loop is similar to the while loop
- With a do-while loop the condition is evaluated at the end of the iteration.
- The loop expressions will be executed at least once

```
do {
    expression;
    expression;
    ...
} while (condition);
```

do-while

Execute the block of code

```java
do {
    System.out.println(i);
    i++;
} while (i <= 1000)
```

Check if i<=1000. If it's true, repeat once more.

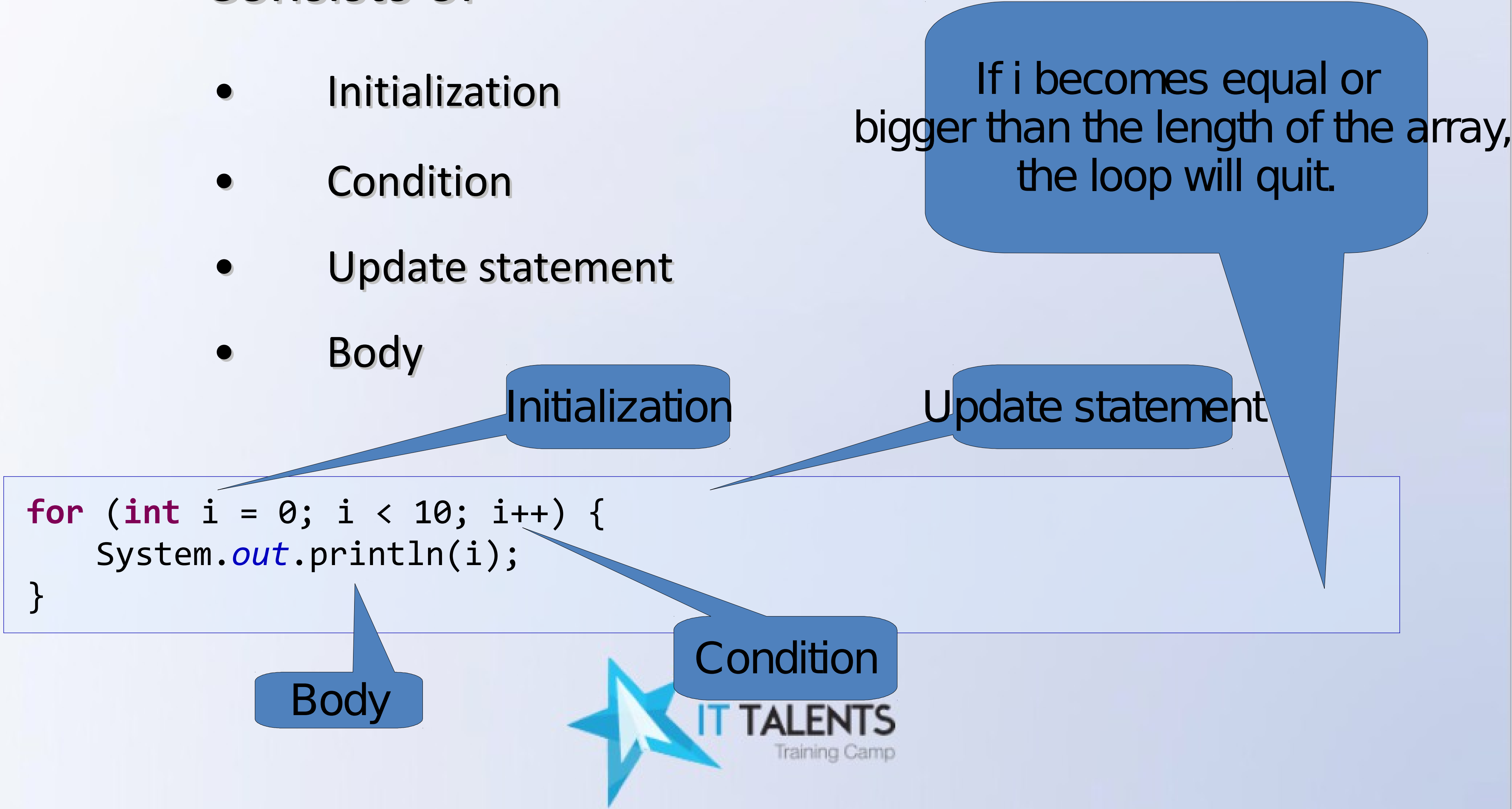
For Loop

- The for loops are another commonly used loop
- There are three important expressions which make the magic

```
for (init_expr; condition_expr; control_expr) {
    expression;
    expression;
    ...
}
```

For loop

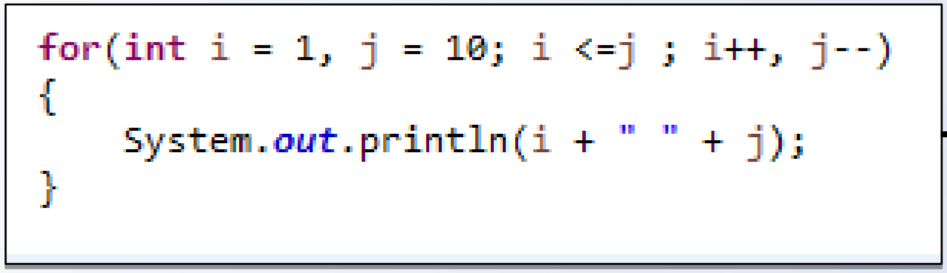
- Consists of
  - Initialization
  - Condition
  - Update statement
  - Body


Initialization
Update statement
If i becomes equal or bigger than the length of the array, the loop will quit.
Condition
Body

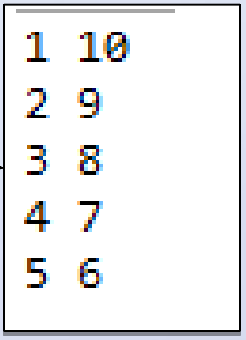
```
for (int i = 0; i < 10; i++) {
    System.out.println(i);
}
```

For loop

- For with more than one variable

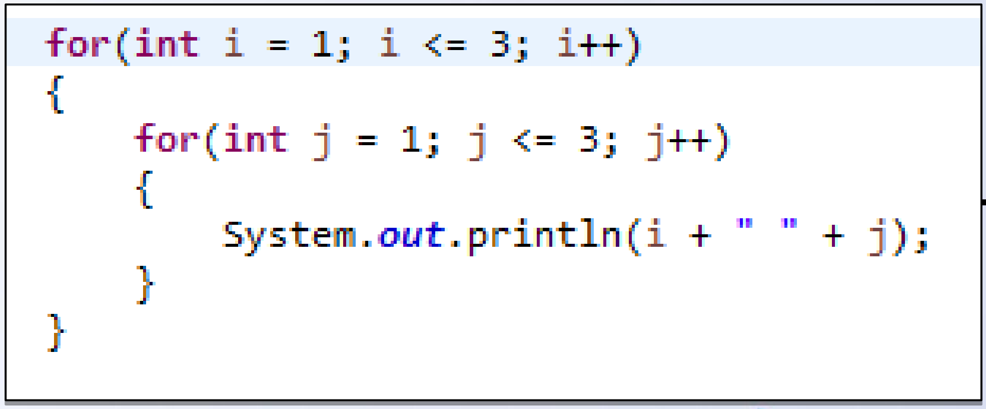


```java
for(int i = 1, j = 10; i <=j ; i++, j--)
{
    System.out.println(i + " " + j);
}
```

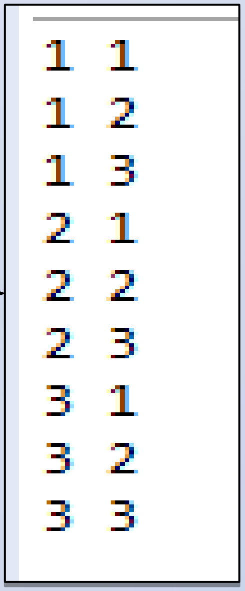


```
1 10
2 9
3 8
4 7
5 6
```

Nested for loops



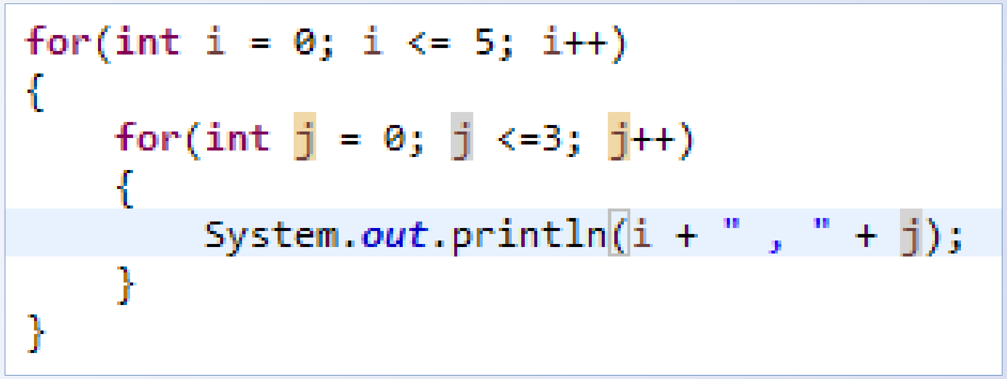
```java
for(int i = 1; i <= 3; i++)
{
    for(int j = 1; j <= 3; j++)
    {
        System.out.println(i + " " + j);
    }
}
```



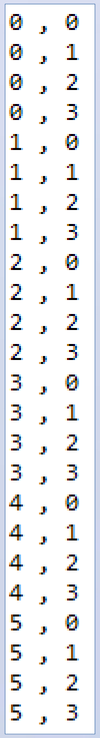
```
1 1
1 2
1 3
2 1
2 2
2 3
3 1
3 2
3 3
```

Nested Loops

- Loops could be nested in each-other
- We can embed loops of different kind
- There is no limit how deep we can go nesting



```
for(int i = 0; i <= 5; i++)
{
    for(int j = 0; j <=3; j++)
    {
        System.out.println(i + " , " + j);
    }
}
```



```
0 , 0
0 , 1
0 , 2
0 , 3
1 , 0
1 , 1
1 , 2
1 , 3
2 , 0
2 , 1
2 , 2
2 , 3
3 , 0
3 , 1
3 , 2
3 , 3
4 , 0
4 , 1
4 , 2
4 , 3
5 , 0
5 , 1
5 , 2
5 , 3
```

Problem

- Try to quit a for-loop during the execution of the repeatable block

- One possible to solution is to set the counter to a value which will make the boolean condition quit the loop....but…

Break

- Break is a keyword
- A statement by itself
- It doesn't require anything else
- It stops the execution of the loop

The loop will quit
when i is equal to

```
for (int i = 0; i < 50; i++) {
    if (i == 7) {
        break;
    }
}
```

IT TALENTS
Training Camp

Problem

- Try to omit specific block of code in the body – for example sum all numbers between 1 and 100 but omit all numbers between 51 and 74
- Encapsulating the code in if-else statements may be used. Although for more complicated structures should be used for more complicated cases

IT TALENTS
Training Camp

Continue

- Continue is a keyword
- A statement by itself
- It doesn't require anything else
- It stops the current iteration of the loop, but doesn't stop the loop

```
for (int i = 0; i < args.length; i++) {
    if (i > 51 && i < 71) {
        continue;
    }
    sum = sum + i;
}
```

If i is between 51 na 74,
the loop will skip
all statements after **continue**

Summary

- Why do we use loops?
- What does a loop consist of?
- Difference between *while* and *do-while*?
- How to use *for* – loop?
- How to terminate a loop?
- How to stop the current iteration?

IT TALENTS
Training Camp