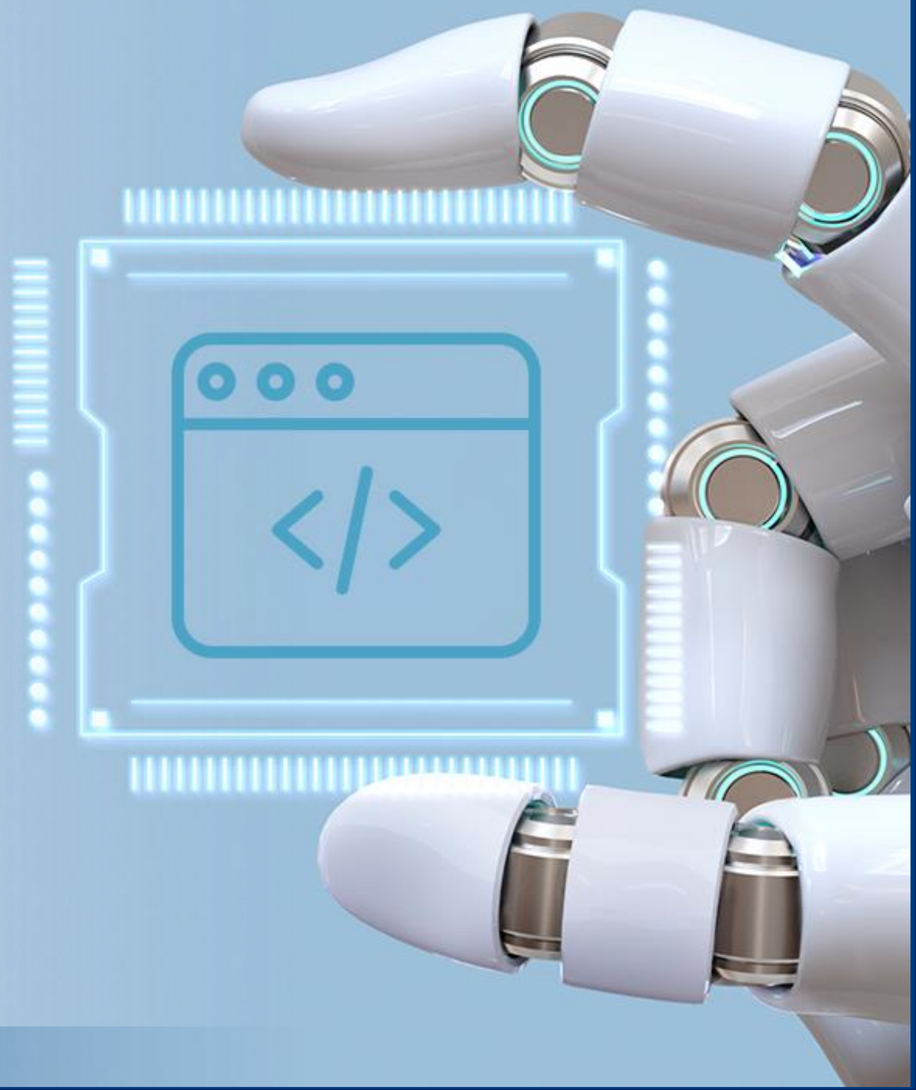
edunet
foundation

Java Control Statements
Java Array & String



Disclaimer

The content is curated from online/offline resources and used for educational purpose only

Agenda

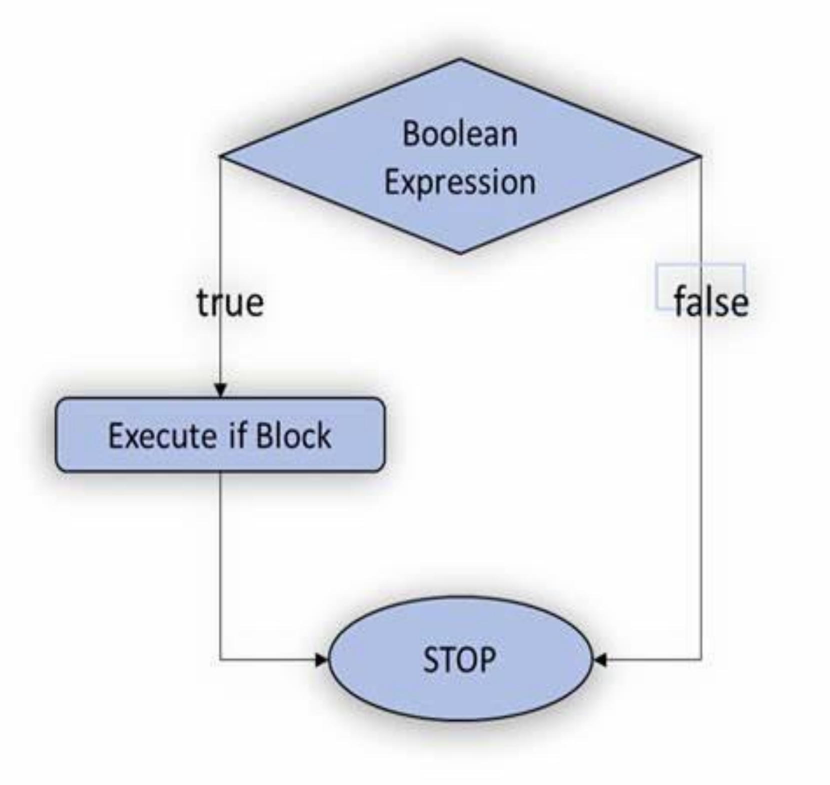
- Understanding Java Control Statements
- Conditional Statements
- Looping Statements
- Break and Continue statements
- Java Array and String

WHY DO WE NEED
CONTROL STATEMENTS?
Apply conditions and use Control statements
Execute statements based on conditions
JavaGoal.com

Source:

Introduction

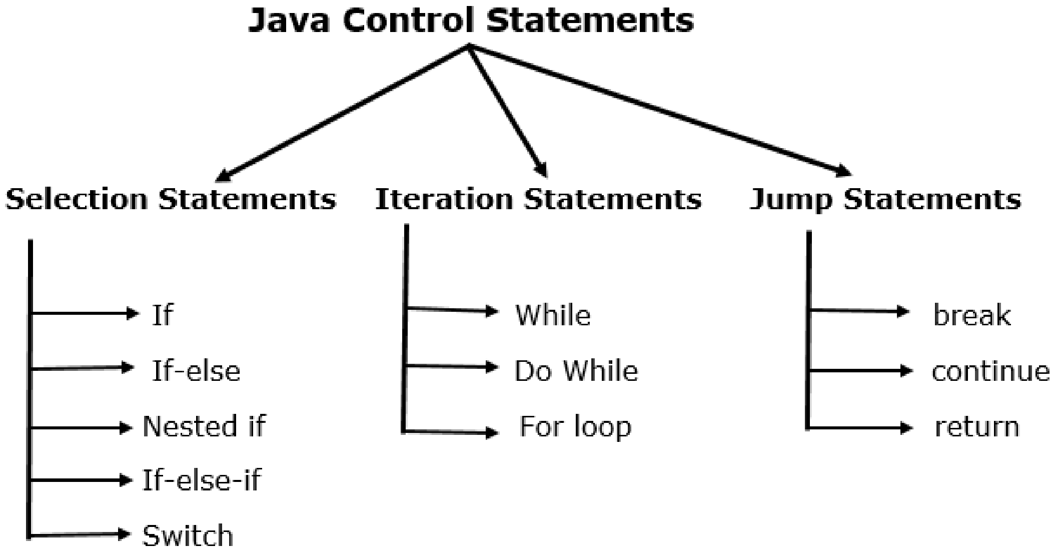
Control statements are fundamental building blocks in programming that allow you to alter the flow of execution in a program based on certain conditions. They enable you to make decisions, repeat blocks of code, and control the logical flow within your program. Control statements provide a way to create dynamic and flexible programs that can respond to different situations.



Source:

Java Control Statements

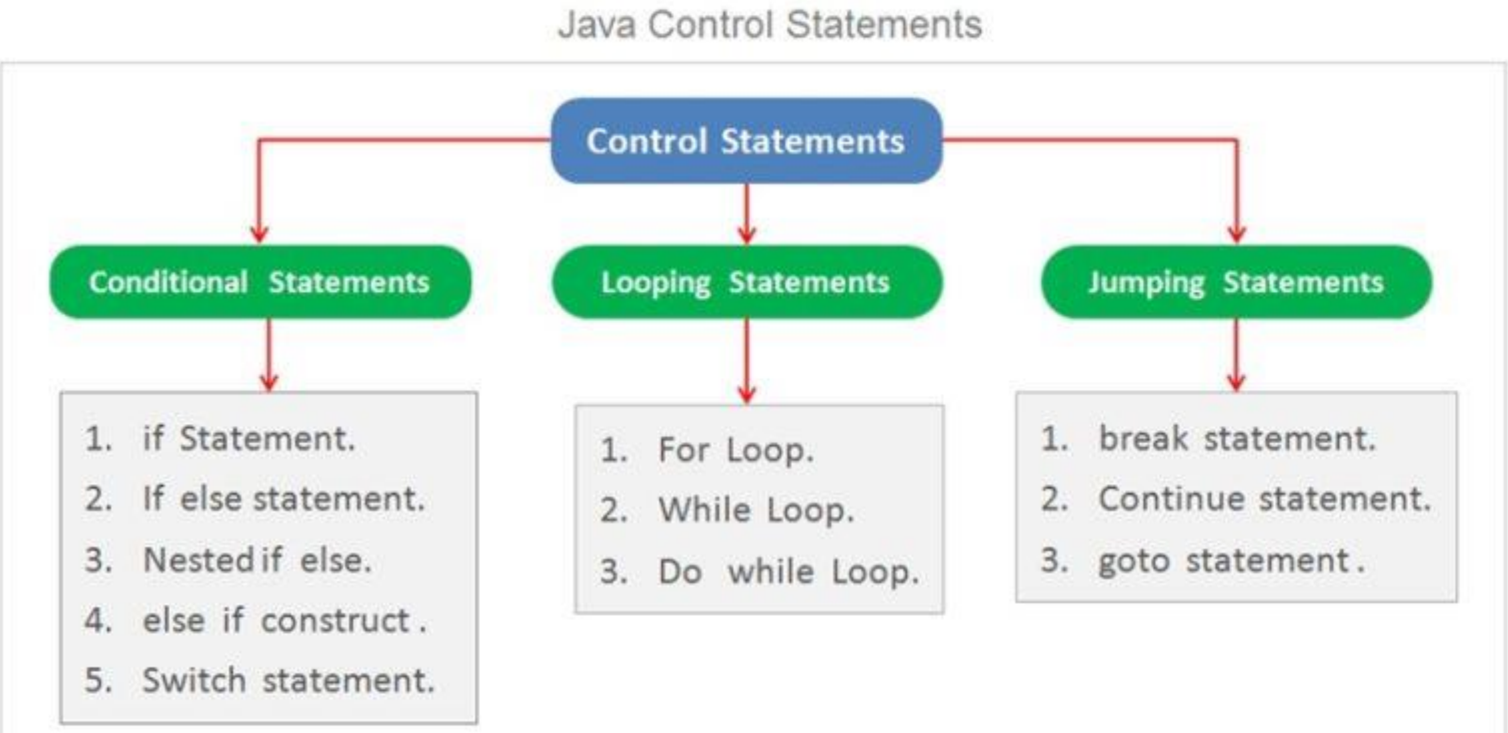
- Control statements are programming constructs that allow you to control the flow of execution in a program.
- They determine which blocks of code are executed and when based on certain conditions or criteria.
- Control statements enable you to make decisions, repeat code, and transfer control to different parts of the program.



Source: https://www.javastudypoint.com/2019/12/java-control-statements.html

Types of Control Statements

1. Conditional Statements
2. Looping Statements
3. Jump Statements



Java Control Statements

Control Statements

Conditional Statements

1. if Statement.
2. If else statement.
3. Nested if else.
4. else if construct .
5. Switch statement.

Looping Statements

1. For Loop.
2. While Loop.
3. Do while Loop.

Jumping Statements

1. break statement.
2. Continue statement.
3. goto statement .

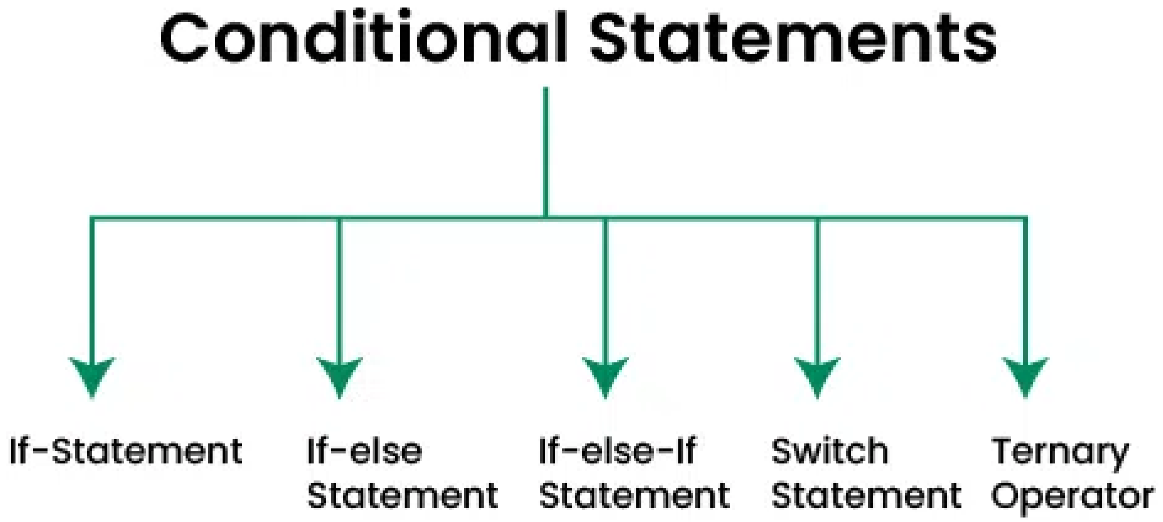
Source: https://www.javastudypoint.com/2019/12/java-control-statements.html

Conditional Statements

- Conditional statements are control statements that allow you to execute specific blocks of code based on certain conditions. They enable your program to make decisions and choose different paths of execution depending on the values of variables or the outcome of comparisons.

Common types of conditional statements:

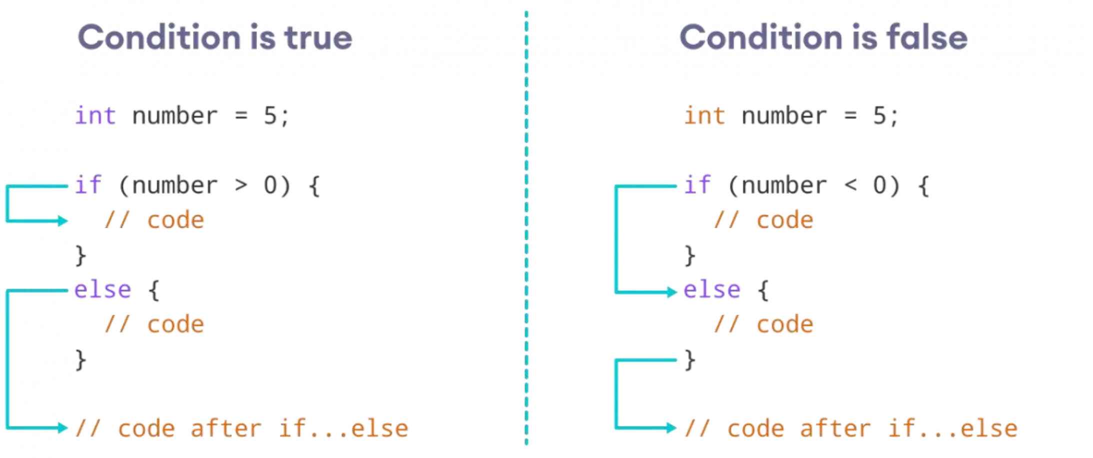
- If
- If-else
- nested



Source: https://www.geeksforgeeks.org/conditional-statements-in-programming/

Java if...else Statement

- We use the if..else statement to run a block of code among more than one alternative.
- The if statement executes a certain section of code if the test expression is evaluated to true.
- However, if the test expression is evaluated too false, it does nothing.
- Statements inside the body of else block are executed if the test expression is evaluated too false.



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
```

How the if...else statement works?

Source: https://www.programiz.com/java-programming/if-else-statement

Lab Exercise

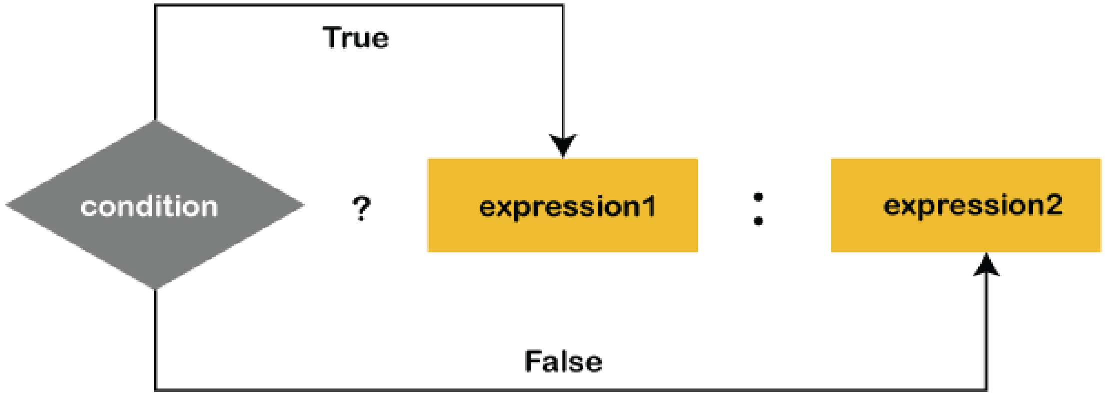
Hands On - 9: Java program that Checks Positive Numbers

Hands On - 10: Java program using the if-else statement to check whether the number is positive or negative

Hands On - 11: Java program using the nested if-else statement to check whether the number is positive or negative or zero

Java Ternary Operator

- In Java, the ternary operator is a type of Java conditional operator.
- The meaning of ternary is composed of three parts.
- The ternary operator (? :) consists of three operands.
- It is used to evaluate Boolean expressions.
- The operator decides which value will be assigned to the variable.
- It is the only conditional operator that accepts three operands.
- It can be used instead of the if-else statement. It makes the code much more easy, readable, and shorter.



Syntax:

variable = (condition) ? expression1 : expression2

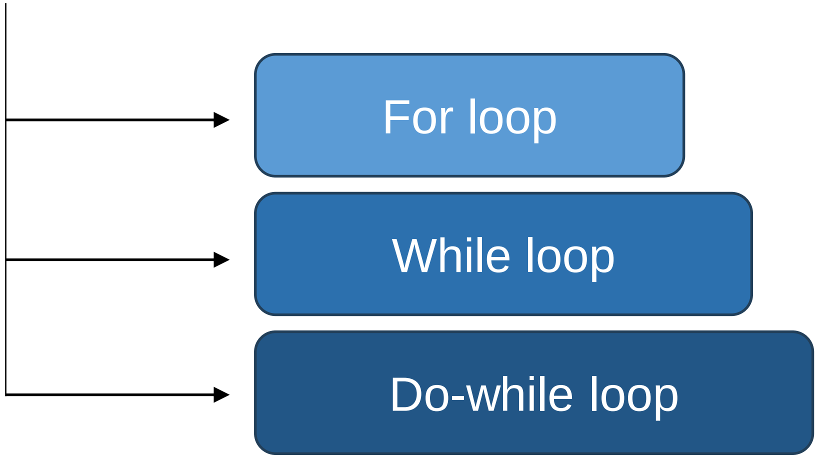
Lab Exercise

Hands On - 12: Example using the ternary operator to determine whether a number is even or odd

Looping Statements

- Looping statements are control statements that allow you to repeat a block of code multiple times. They are used when you want to perform a certain task repeatedly until a specific condition is met. Loops provide a way to automate repetitive tasks, iterate over collections of data, and ensure efficient processing.

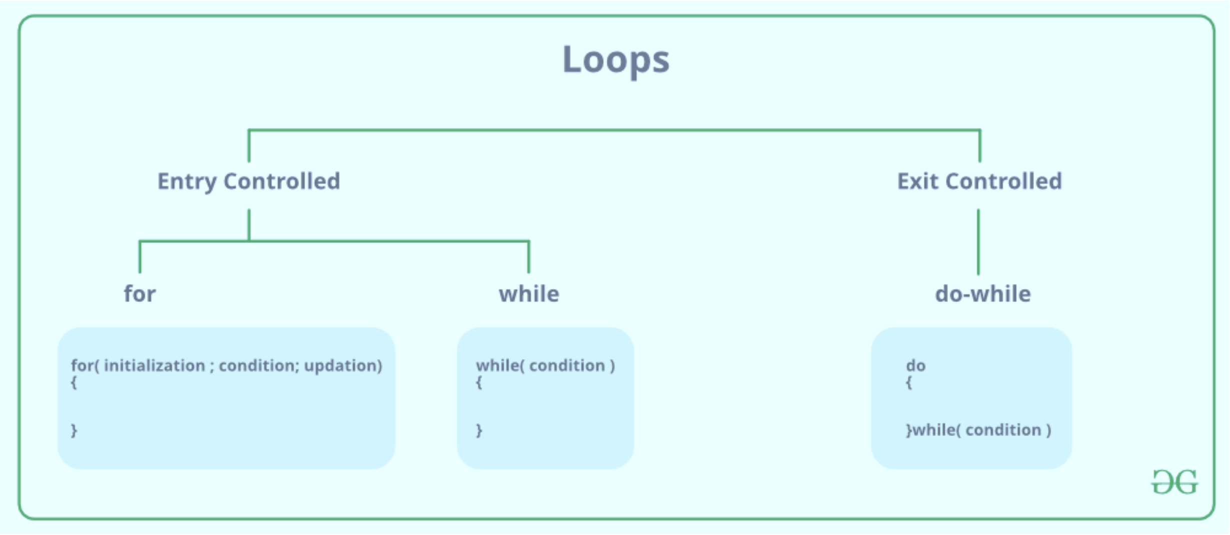
Common types of looping statements:



For loop

While loop

Do-while loop



```
for( initialization ; condition; updation)
{

}
```

```
while( condition )
{

}
```

```
do
{

}while( condition )
```

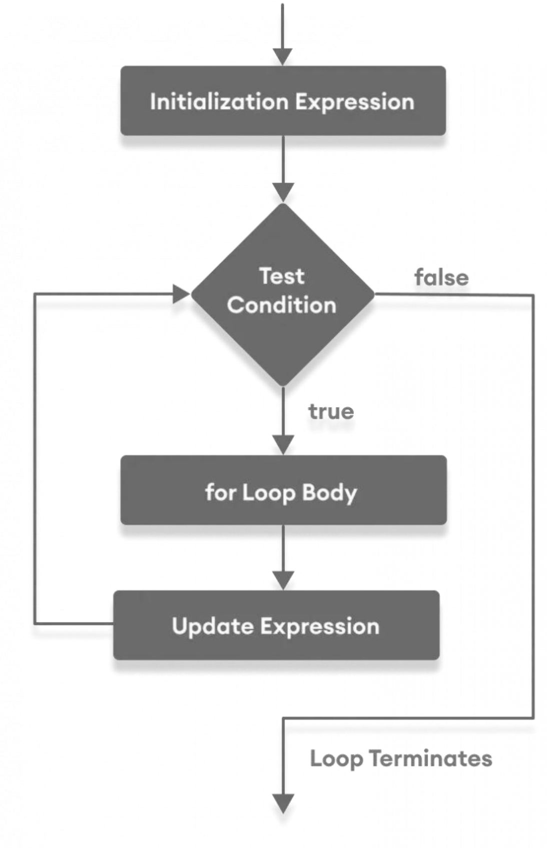
Source: https://www.geeksforgeeks.org/c-loops/

Java for Loop

- In computer programming, loops are used to repeat a block of code.
- Java for loop is used to run a block of code for a certain number of times.

Syntax

```
            for (initialization; condition;
increment/decrement) {
   // Code to execute in each iteration
            }
```



Flowchart of Java for loop

Lab Exercise

Hands On - 13: Example of using the for loop

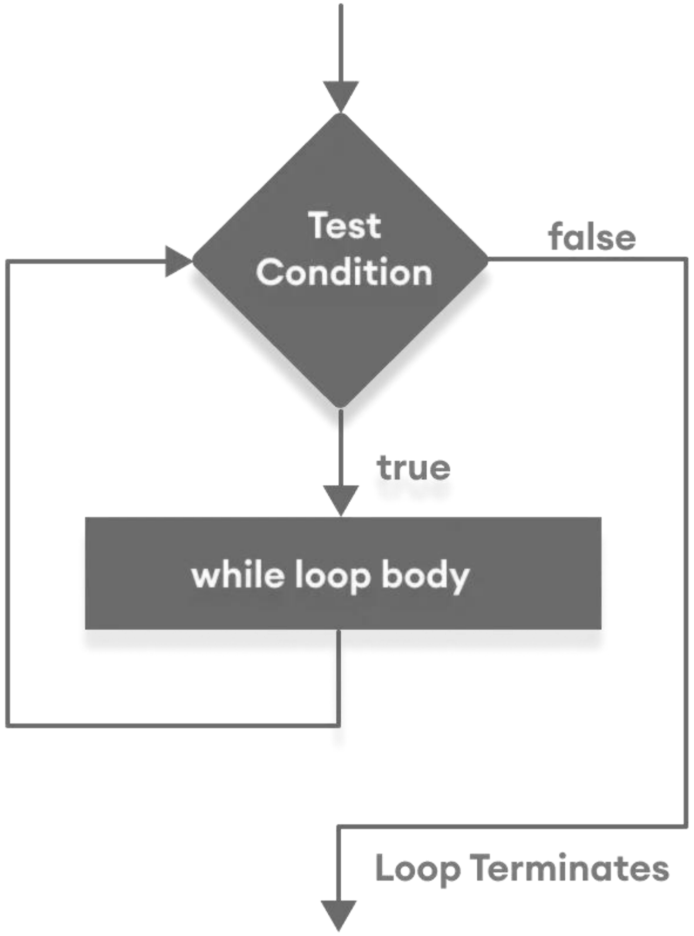
- To print numbers from 1 to 5.
- To calculate the sum of numbers from 1 to 10.

Java while loop

- Java while loop is used to run a specific code until a certain condition is met.

Syntax

```
            while (condition) {
   // Code to execute in each iteration
}
```



Flowchart of while loop

Lab Exercise

Hands On - 14: Example of using the while loop

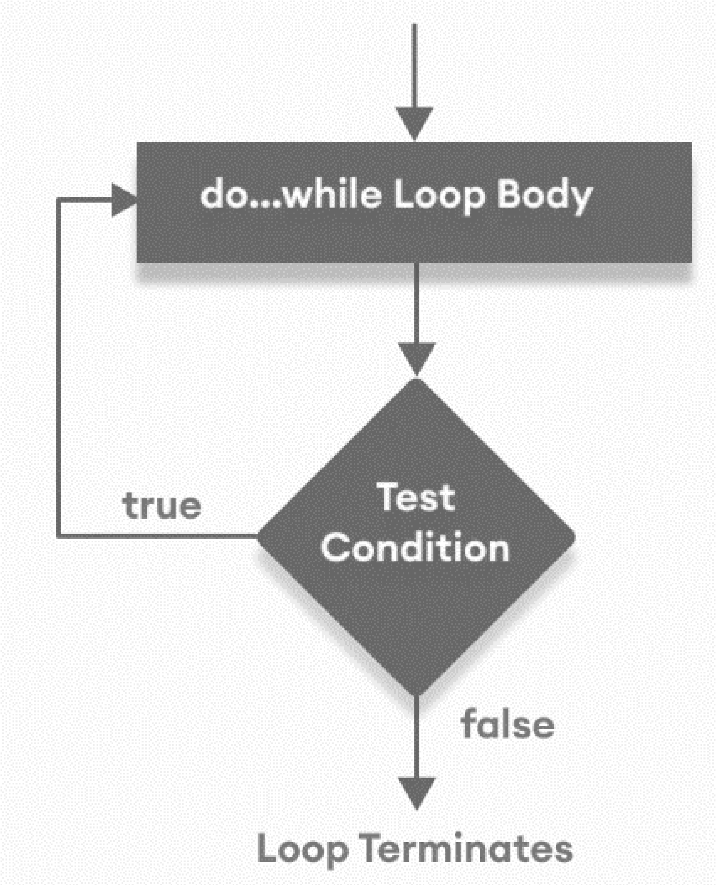
- To print numbers from 1 to 5.
- To calculate the sum of numbers from 1 to 10.

Java do-while loop

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

Syntax

```
        do {
  // Code to execute in each iteration
 } while (condition);
```

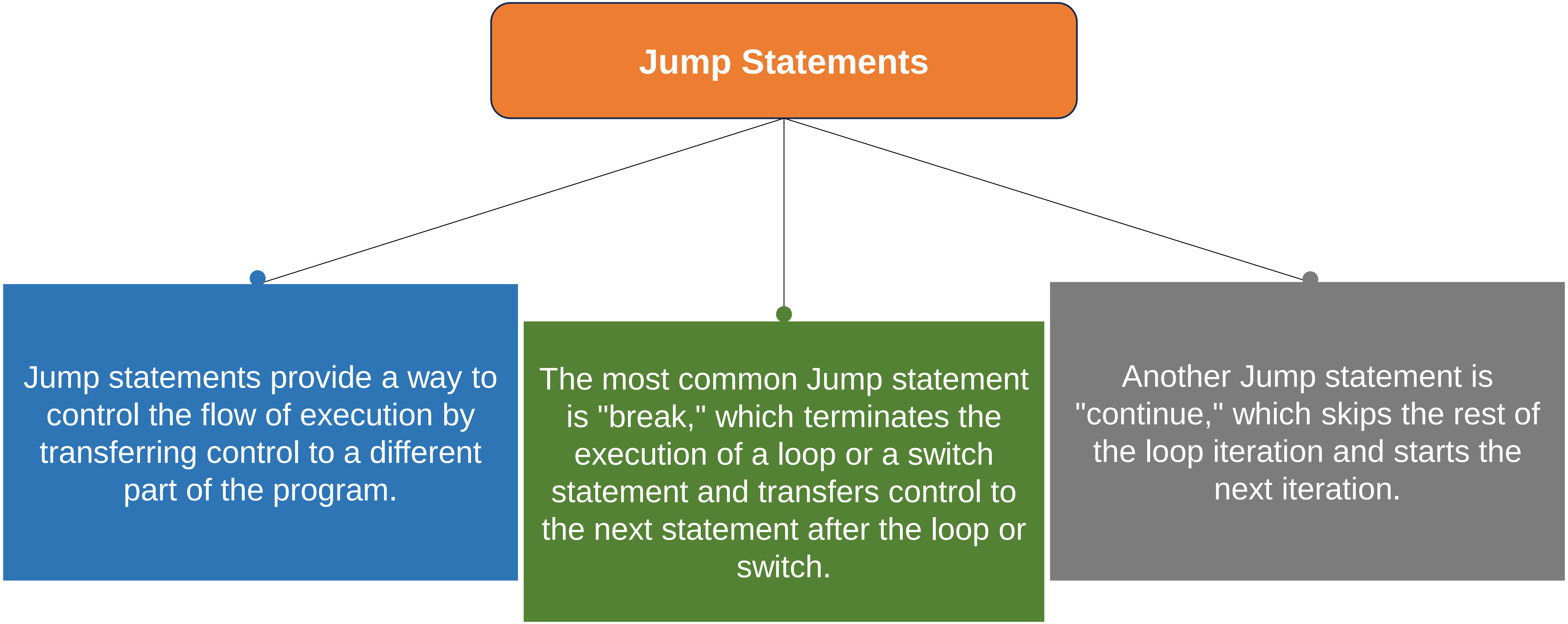


Flowchart of do-while loop

Lab Exercise

Hands On - 15: Example of using the do while loop

- To print numbers from 1 to 5.
- To calculate the sum of numbers from 1 to 10.



Jump Statements

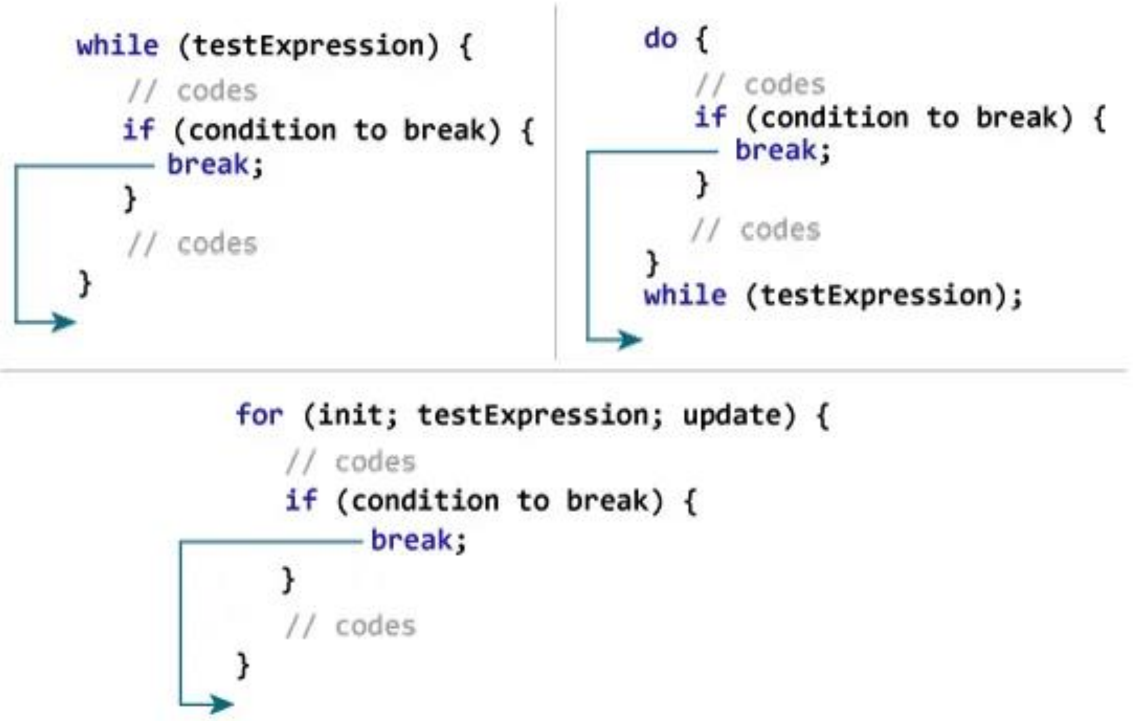
Jump statements provide a way to control the flow of execution by transferring control to a different part of the program.

The most common Jump statement is "break," which terminates the execution of a loop or a switch statement and transfers control to the next statement after the loop or switch.

Another Jump statement is "continue," which skips the rest of the loop iteration and starts the next iteration.

Java break Statement

- While working with loops, it is sometimes desirable to skip some statements inside the loop or terminate the loop immediately without checking the test expression.
- 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 (Java if...else Statement).



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

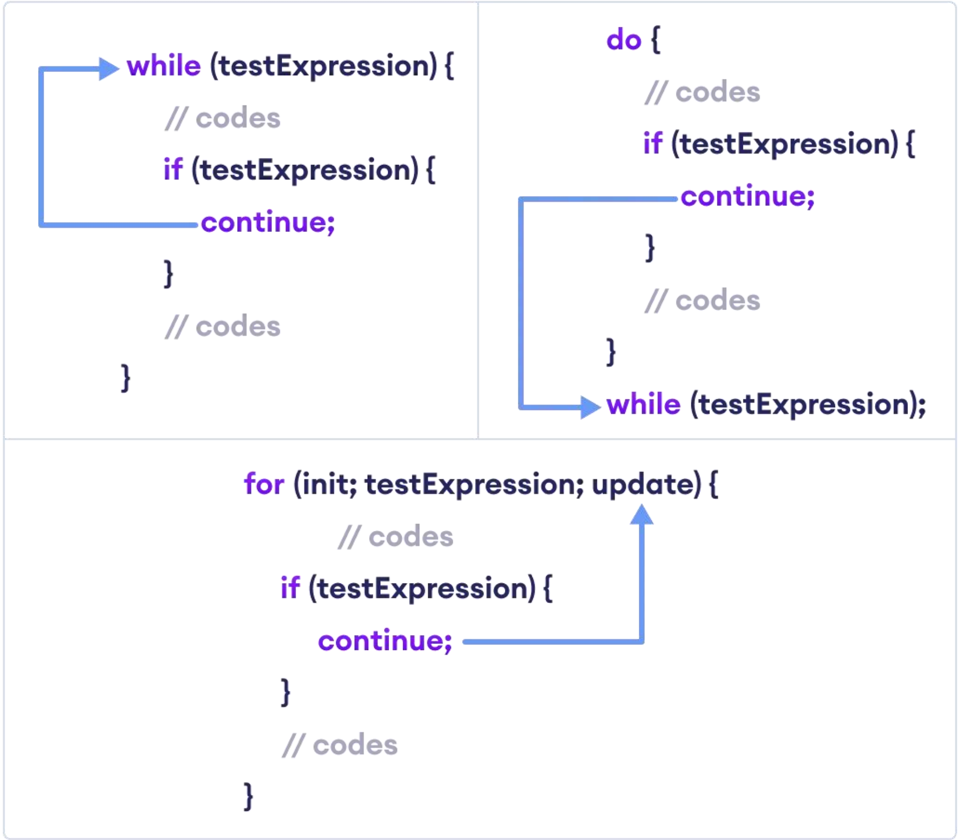
```
do {
    // codes
    if (condition to break) {
        break;
    }
    // codes
}
while (testExpression);
```

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

Working of Java break Statement

Java 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).



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

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

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

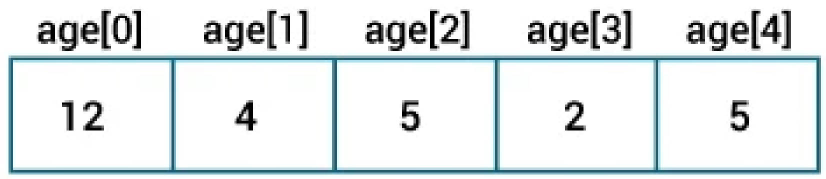
Working of Java continue Statement

Java Arrays

- An array is a collection of similar types of data.
- Array indices always start from 0. That is, the first element of an array is at index 0.
- If the size of an array is n, then the last element of the array will be at index n-1.

Syntax:

```
arrayName = new datatype[size];
```

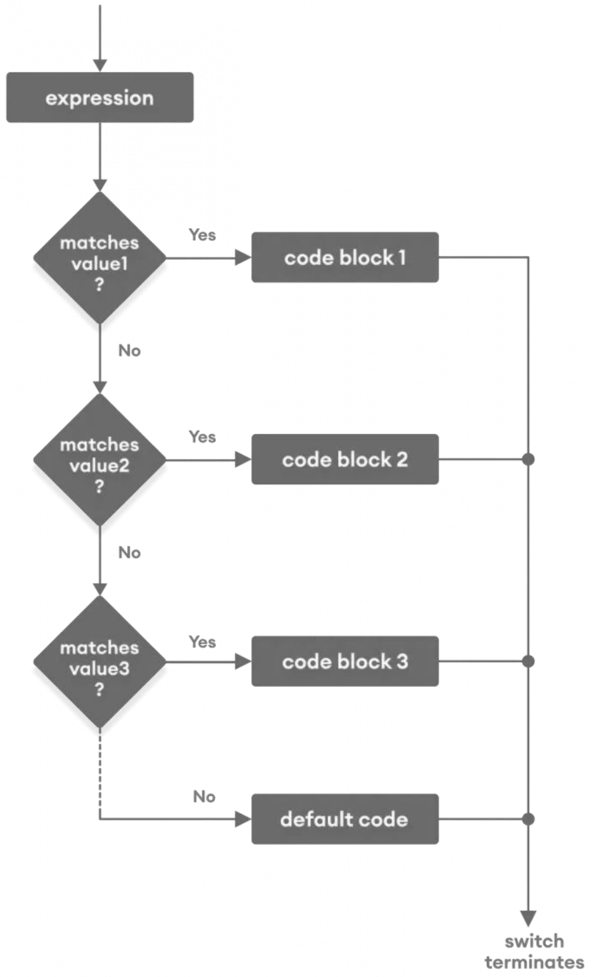


age[0]	age[1]	age[2]	age[3]	age[4]
12	4	5	2	5

Java Arrays initialization

Java switch Statement

- The switch statement allows us to execute a block of code among many alternatives.
- The expression is evaluated once and compared with the values of each case.
- The working of the switch-case statement is like the Java if...else...if ladder.
- However, the syntax of the switch statement is cleaner and much easier to read and write.



Flowchart of switch Statement

Lab Exercise

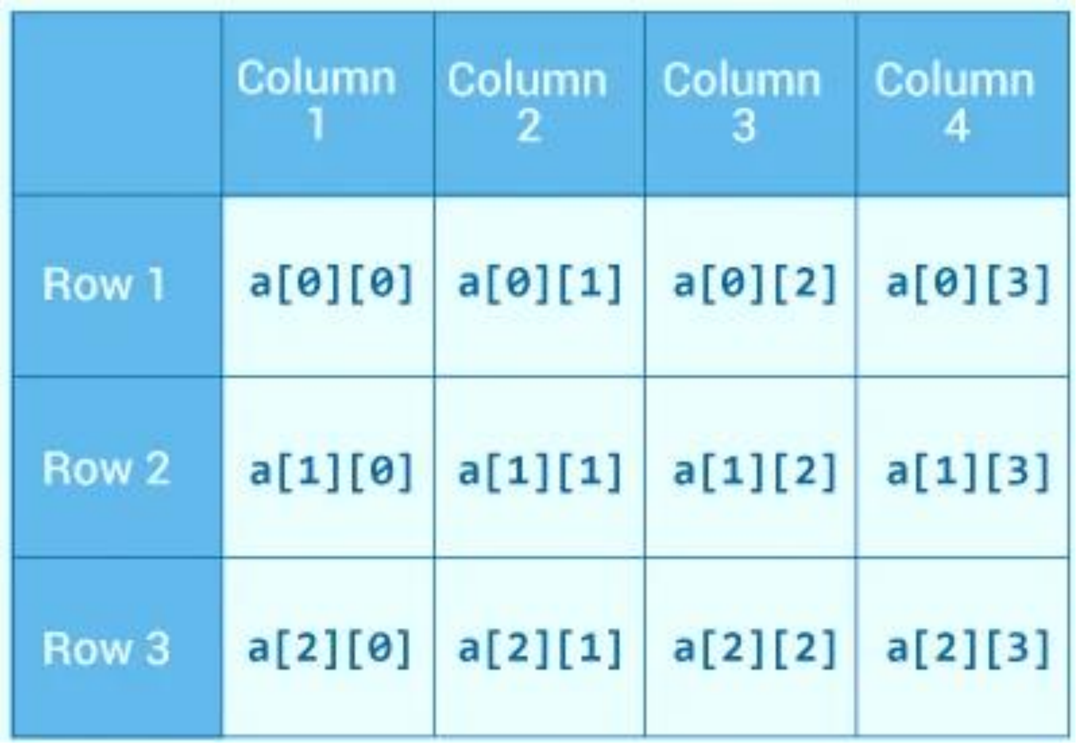
Hands On - 16: Java program using the switch statement to determine the name of a day based on its number (1 to 7)

Hands On - 17: Java program to terminate for loop using a break statement

Hands On - 18: Using the 'continue' Statement to Skip Loop Iterations

Java Multidimensional Arrays

- A multidimensional array is an array of arrays.
- Each element of a multidimensional array is an array itself.
- Each element of the multidimensional array is an array itself.
- unlike C/C++, each row of the multidimensional array in Java can be of different lengths.



	Column 1	Column 2	Column 3	Column 4
Row 1	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 2	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 3	a[2][0]	a[2][1]	a[2][2]	a[2][3]

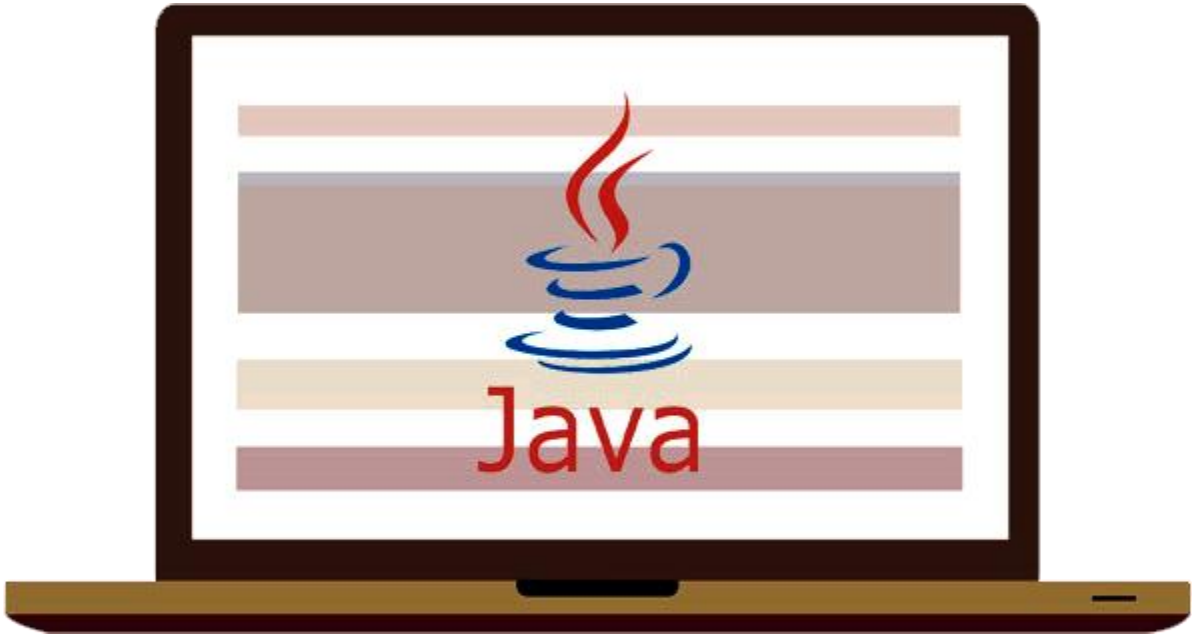
2-dimensional Array

Lab Exercise

Hands On - 19: Example of how to create and initialize an integer array with some values

Java String

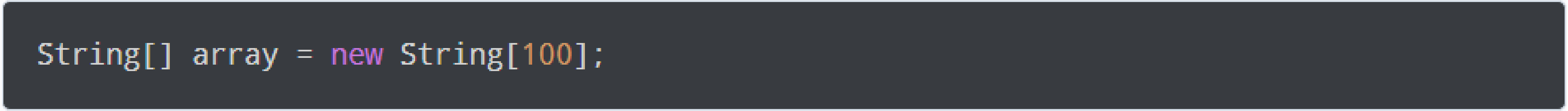
- Strings are a fundamental data type in programming used to represent sequences of characters. In most programming languages, strings are typically enclosed in single quotes ('') or double quotes ("").
- String methods, also known as string functions or string operations, are built-in functions provided by programming languages that allow you to manipulate and perform operations on strings. These methods help you work with strings more effectively by providing functionalities such as string concatenation, searching for substrings, modifying case, splitting and joining strings, and much more.



String Methods

Here are some commonly used string methods:

- **Concatenation:** - concat()
- **Length and Access:** - length(), substring()
- **Searching and Comparing:** - indexOf(), equals()
- **Splitting and Joining:** - split(), join()
- **Modifying and Replacing:** - replace(), trim()



```
String[] array = new String[100];
```

Java Arrays initialization

Lab Exercise

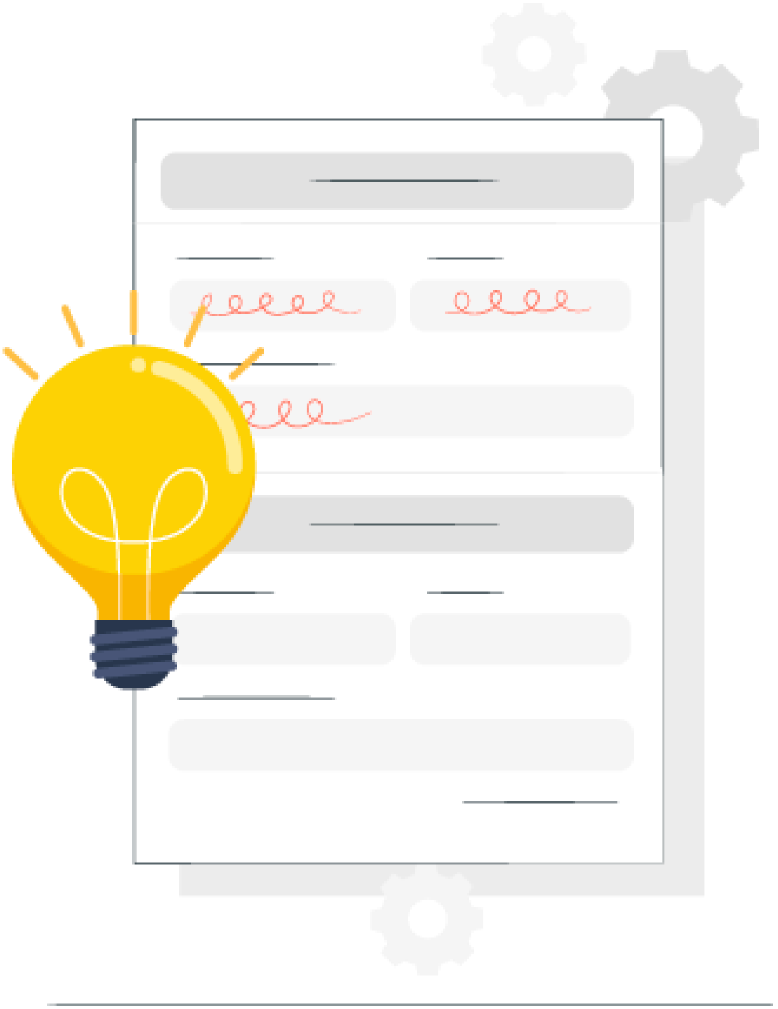
Hands On - 20: Example that demonstrates the use of the String methods

Hands On - 21: Example the demonstrates use of String methods like concat(), length(), substring(), indexOf(), join(), split(), replace()

Conclusion

Well done! You have completed this course and now you understand about:

- Java Control Statement
- Types of Control Statements
- Conditional Statements
- Looping Statements
- Jump Statements
- Java array
- Java string and String Methods.



Quiz

1. Which loop construct in Java is best suited when the number of iterations is known?

a) for loop

b) while loop

c) do-while loop

d) break statement

Answer: a
for loop

Quiz

2. Which of the following jump statements is used to terminate the current iteration of a loop in Java?

a) break

b) continue

c) return

d) Exit

Answer: a
break

edunet
foundation

Quiz

3. Which of the following loop statements in Java guarantees that the loop body will execute at least once?

a) while loop

b) for loop

c) do-while loop

d) None of the above

Answer: c
do-while loop

Quiz

4. What is the output of the following code?

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

a) 5

b) 0

c) 4

d) Compilation Error

Answer: a
5

Quiz

5. How do you declare an array in Java?

a) int[] arr;

b) int arr[];

c) int arr;

d) Array<int> arr;

Answer: : a and b is correct

References

- https://www.javaguides.net/2021
- https://www.geeksforgeeks.org/java/
- https://www.w3schools.com/java/java_break.asp
- https://www.javatpoint.com/control-flow-in-java
- https://www.youtube.com/watch?v=XHgC6Md8L9o

Thank You!