Day 2

Basic Java

Please mute your microphone.

If you have any questions during the session type your questions in chat box

Operator

- Less than: a < b
- Less than or equal to: a <= b
- Greater than: a > b
- Greater than or equal to: a >= b
- Equal to a == b
- Not Equal to: a != b

Ternary Operator

variable x = (expression) ? if true: if false

```
String ternaryOperator = (24 < 18) ? "Greater":"Smaller";
System.out.println(ternaryOperator);
```

Java If ... Else

The if Statement Syntax :

```
if (condition) {
  // block of code to be executed if the condition is true
}
```

The else Statement Syntax :

```
if (condition) {
  // block of code to be executed if the condition is true
} else {
  // block of code to be executed if the condition is false
}
```

Java If ... Else

Java has the following conditional statements:

- If is used to specify a block of code to be executed, if a specified condition is true.
- Else is used to specify a block of code to be executed, if the same condition is false.
- Else if is used to specify a new condition to test, if the first condition is false.
- Switch is used to specify many alternative blocks of code to be executed.

String

length()
contains()
equals()
equlIgnoreCase()
endsWith()
startsWith()
trim()

indexOf()
lastIndexOf()
replace()
replaceAll()
substring()
split()
toCharArray()
ValueOf()

If ... Else

The else if Statement Syntax :

```
if (condition1) {
  // block of code to be executed if condition1 is true
} else if (condition2) {
  // block of code to be executed if the condition1 is false and condition2 is true
} else {
  // block of code to be executed if the condition1 is false and condition2 is false
}
```

Switch

Switch Statements Syntax :

Switch is used to specify many alternative blocks of code to be executed.

```
switch(expression) {
  case x:
    // code block
    break;
  case y:
    // code block
    break;
  default:
    // code block
}
```

Java Switch

How it works:

- The switch expression is evaluated once.
- The value of the expression is compared with the values of each case.
- If there is a match, the associated block of code is executed.
- The break and default keywords are optional

For Loop

For Loop Syntax :

When we know exactly how many times we want to loop through a block of code, use the for loop instead of a while loop:

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
 }
```

Statement 1 is executed (one time) before the execution of the code block.

Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.

For Loop

The example below will print the numbers 0 to 4:

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

Example explained

Statement 1 sets a variable before the loop starts (int i = 0).

Statement 2 defines the condition for the loop to run (i must be less than 5). If the condition is true, the loop will start over again, if it is false, the loop will end.

Statement 3 increases a value (i++) each time the code block in the loop has been executed.

Break

Java Break

You have already seen the break statement used in an earlier chapter of this tutorial. It was used to "jump out" of a switch statement.

The break statement can also be used to jump out of a loop.

This example jumps out of the loop when i is equal to 4:

```
for (int i = 0; i < 10; i++) {
  if (i == 4) {
    break;
  }
  System.out.println(i);
}
```

Continue

Java Continue

The continue statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

This example skips the value of 4:

```
for (int i = 0; i < 10; i++) {
  if (i == 4) {
    continue;
  }
  System.out.println(i);
}
```

While Loop

Loops :

- Loops can execute a block of code as long as a specified condition is reached.
- Loops are handy because they save time, reduce errors, and they make code more readable.

While Loop Syntax

The while loop loops through a block of code as long as a specified condition is true:

```
while (condition) {
  // code block to be executed
}
```

Arrays

Java Arrays

Arrays are used to store multiple values in a single variable, instead of declaring separate variables for each value.

To declare an array, define the variable type with square brackets: