Loops

while, do while, for

Learning objectives

- The while loop
- The do while loop
- The for loop
- The break and continue statements
- Nested loops

The while loop

- Repeats a block of code as long as a condition evaluates to true
- Ends the repetition when/if the condition evaluates to false
- Repeats the block of code from zero to an infinite number of times

Demo 1 - The while loop

- A use case for the while loop
- Iteration zero to an infinite number of times

The do while loop

- Runs a block of code, and repeats it as long as a condition evaluates to true
- Ends the repetition when/if the condition evaluates to false
- Repeats the block of code from one to an infinite number of times

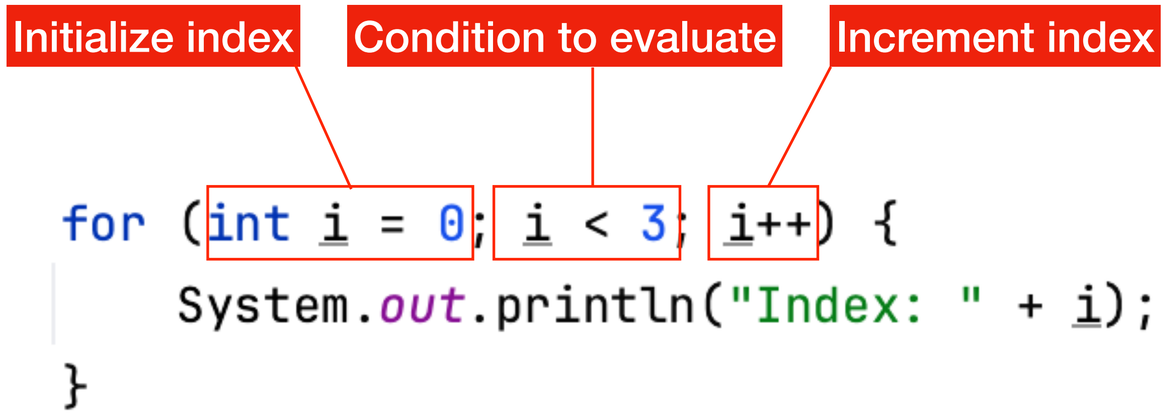
Demo 2 - The do while loop

- A use case for the do while loop
- Iteration one to an infinite number of times

The for loop

- The for loop repeats a code block a specified number of times
- It is controlled by an index
- A condition is evaluated before every iteration
- The code block is repeated as long as the condition evaluates to true

The for loop



- Very flexible - the index can be initialized to anything, the condition could evaluate anything and the increment (or decrement) of the index can be anything

Demo 3 - The for loop

- Examples of different for loops
- Initializing the index, the condition, the increment/decrement

Exercise 1 - while and do while

- Make a copy of the Calculator solution
- This time do not accept an invalid operator as input
- Validate the input for the operator and if it is not a valid operator then ask the user again for a valid operator. Do not continue with the calculation until a valid operator is entered
- Create one solution using a while loop, and one using a do while loop

Exercise 2 - do while

- Make a copy of the last Calculator solution
- Make it possible to calculate many numbers (the program should not end after one calculation)
- After a calculation, ask if the user wants to do another calculation by entering y for yes or quit by entering n for no
- Read the input as a String with the scanner.next() method and repeat the calculation as long as the user enters y

Exercise 3 - for

- Task 1 - 4: Create for loops that print the following output:

```
Task 1: 1 2 3 4 5
Task 2: 8 9 10 11 12
Task 3: 12 11 10 9 8
Task 4: 10 20 30 40 50
```

Exercise 4 - FizzBuzz

- Create a for loop that prints numbers from 1 to 100
- But for multiples of three print “Fizz” instead of the number
- And for multiples of five print “Buzz”
- And for numbers which are multiples of both three and five print “FizzBuzz”

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
```

break and continue

- break - breaks out of the entire loop, ends the loop
- continue - continues the loop, breaks out of the iteration, ends the iteration but not the loop

Demo 4 - break and continue

- Using break
- Using continue

while (true) without break;

ACTLEA
Active Learning

Nested loops

- Loops within loops
- Repeating sub tasks within a repeating main task

Demo 5 - Nested loops

- Nested for loops

Exercise 5 - break

- Make a copy of the last Calculator solution that uses one loop for entering a valid operator and one loop for repeating the calculation
- Change both loops to infinite while loops by just having the value true as condition, like this: while(true)
- Then use the break keyword inside an if statement to break out of the loop when a valid operator is entered/if the user answers no to another calculation

Exercise 6.1 - Nested loops

- Use nested loops to create a program that prints output like this:

```
12345
12345
12345
12345
12345
```

ACTLEA
Active Learning

Exercise 6.2 - Nested loops

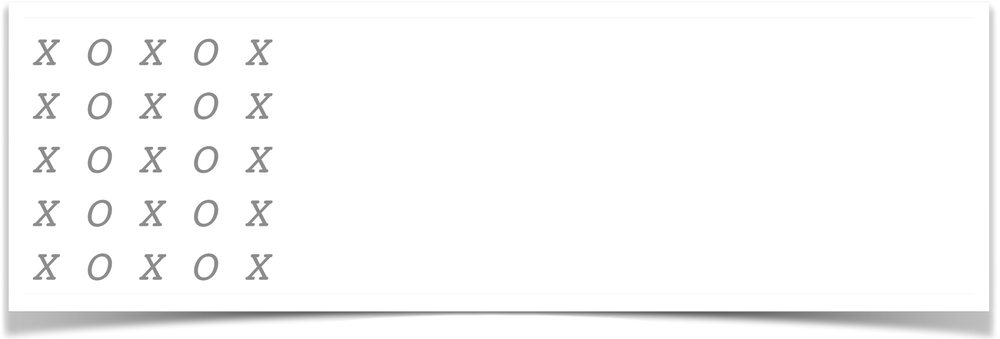
- Use nested loops to create a program that prints output like this:

```
[1-1] [1-2] [1-3] [1-4] [1-5]
[2-1] [2-2] [2-3] [2-4] [2-5]
[3-1] [3-2] [3-3] [3-4] [3-5]
[4-1] [4-2] [4-3] [4-4] [4-5]
[5-1] [5-2] [5-3] [5-4] [5-5]
```

Exercise 6.3 - Nested loops

- Use nested loops to create programs that print output like this:

- The difference between the two programs could be only one character



Exercise 6.4 - Nested loops

- Use nested loops to create a program that creates output like this:

```
X O X O X
O X O X O
X O X O X
O X O X O
X O X O X
```

Exercise 6.5 - Nested loops

- Stretch Task!

- Use nested loops to create a program that creates output like this:

```
1****
12***
123**
1234*
12345
```

ACTLEA
Active Learning

Summary

ACTLEA
Active Learning

Learning objectives

- The while loop
- The do while loop
- The for loop
- The break and continue statements
- Nested loops