Arrays and for loops

Camille Duquesne
1ère IB

Any questions/difficulties you would like to share about last session's content ?

B2 Programming

B2.2.2 Construct programs that apply arrays and Lists.

B2.3.3 Construct programs that utilize looping structures to perform repeated actions.

- Types of loops, including counted loops and conditional loops, and appropriate use of each type
- Conditional statements within loops, using Boolean and/or relational operators to govern the loop's execution

B2.3.1 Construct programs that implement the correct sequence of code instructions to meet program objectives.

- The impact of instruction order on program functionality
- Ways to avoid errors, such as infinite loops, deadlock, incorrect output

What is an Array ?

Why are arrays useful ? Can't I just stick with individual variables ?

An array is a group of elements of the same type. An array has a fixed size, you cannot add or remove elements.

There are two ways of instantiating an array:

Without knowing the content

```
int[] arr = new int[5];
```

brackets

Size of array in brackets

Here, default content will be 0

Knowing the content

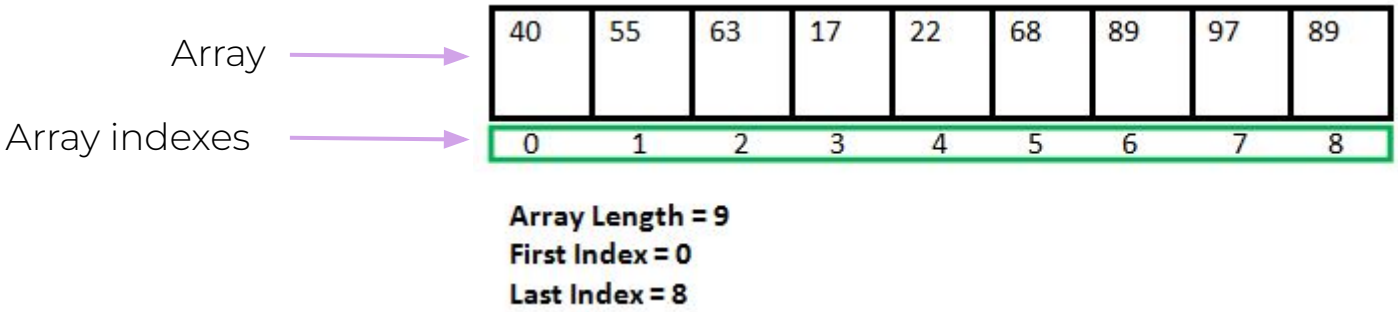
```
int[] arr = { 1,2,3,4,5 };
```

brackets

Elements of the array in curly brackets

Indexes of an array

Array elements can be accessed through their index values. As in most programming languages indexes begin at 0.



How do I access an element of an array ?

```
int[] arr = { 16,23,31,34,15 };
System.out.println(arr[2]);
```

Array variable name

Index between brackets

What if we want to see all of our array elements ?

```
int[] arr = { 16,23,31,34,15 };
System.out.println(arr);
```

```
[I@4617c264
```

This happens because the variable only stores the location of the content in memory. To read the content in memory we use the toString() methods on our array.

What if we want to see all of our array elements ?

```
int[] arr = { 16,23,31,34,15 };
System.out.println(Arrays.toString(arr));
```

```
[16, 23, 31, 34, 15]
```

The toString() method converts each element of the array to a string, separates them with commas, then surrounds them with brackets

Static data structure

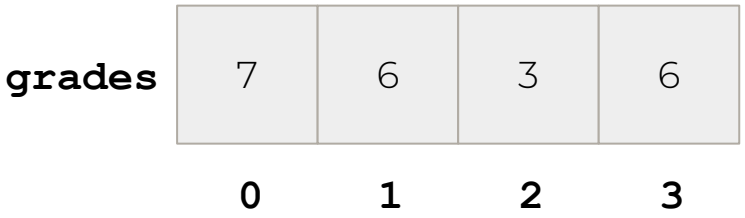
An array is what we call a **static data structure.** This means it has:

- A predefined fixed size upon creation
- A specific memory space being allocated to it
- Elements of the data structure are stored in contiguous memory locations

Elements in a static data structure can either be **directly accessed** (through their index) or can be **sequentially accessed** (through a loop)

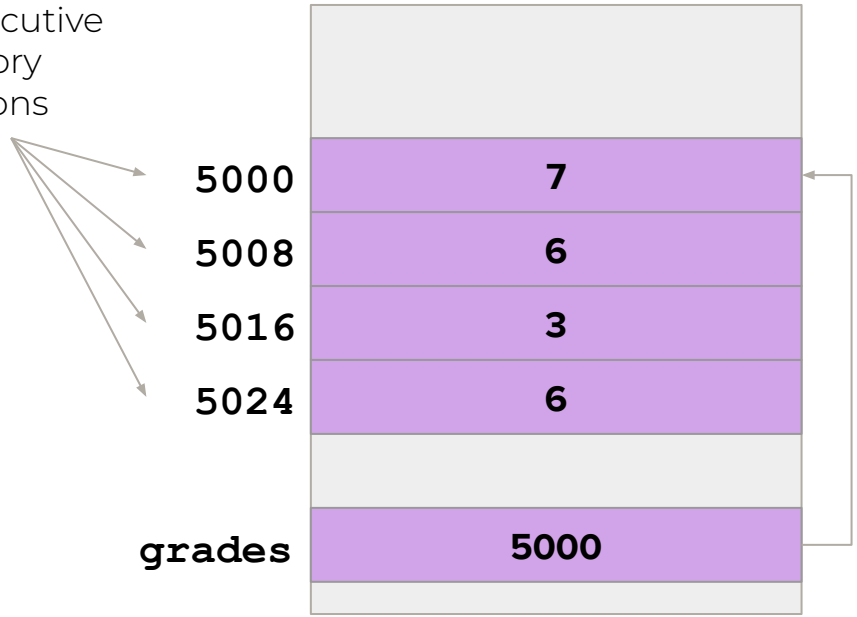
Static data structures and memory

How we perceive an array



Random Access Memory

Consecutive memory locations



Advantages of static data structures

- Fast access time -> through index access
- No memory overflow (or underflow possible) -> size is fixed
- Easier to program -> size is always know and doesn’t need to be computed

Disadvantages of static data structures

- Inefficient use of memory -> since size is fixed, bigger arrays are often created to prevent running out of space
- Memory is allocated (and reserved) at the creation of the array -> whether or not it is used
- No flexibility -> since size is fixed
- Insertion or deletion of an element is cumbersome

For loops

Why are loops useful ?

for loop *(or count controlled loop)* allows us to create a finite number of loops where a block of code will be reused. The syntax is following:

```
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 loops example

If we want to print out each element of the array we would do the following:

```
int[] grades = { 16,23,31,34,15 };
for (int i = 0; i < grades.length; i++) {
    System.out.println(grades[i]);
}
```

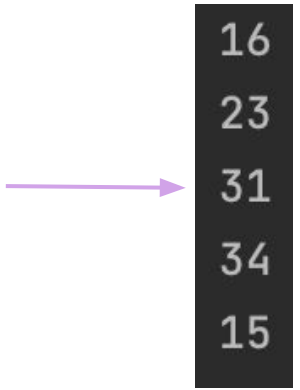
What do we need to change in order to print our grades in % format ?

For each loop

The for each loop is also a type of for loop that is exclusively used for Arrays.

This loop directly takes the values of the array during each iteration.

```
int[] grades = { 16,23,31,34,15 };
for (int grade : grades) {
    System.out.println(grade);
}
```



When indexes are not needed, having a “for each loop” is preferred to having a “for loop”

Introducing trace table

A **trace table** is a technique used to test an algorithm and predict **step by step** how the computer will run the algorithm. It can be used to understand or predict what an algorithm is doing and to identify potential logic errors (when the program compiles but does not produce the expected output).

Using a trace table to test an algorithm is called **dry run testing**.

Algorithm

```
1  number = 3
2  PRINT number
3  FOR i from 1 to 3:
4      number = number + 5
5      PRINT number
6  PRINT " ? "
```

Trace Table

Line	number	i	OUTPUT

What does this code output ?

```java
public class Main {
    public static void main(String[] args) {
        char[] chars = {'a', 'b', 'c', 'd'};
        System.out.println(chars[0]);
    }
}
```

What does this code output ?

```java
public static void main(String[] args) {
    char[] chars = {'a', 'b', 'c', 'd'};
    System.out.println(chars[4]);
}
```

What does this code output ?

```java
public class Main {
    public static void main(String[] args) {
        double[] nums = {3.26, 6.5, 8.12 };
        for (int a = 0; a < nums.length; a++) {
            System.out.println(nums[a]);
        }
    }
}
```

What does this code output ?

```java
public class Main {
    public static void main(String[] args) {
        double[] nums = {3.26, 6.5, 8.12 };
        for (int a = 0; a < nums.length; a++) {
            System.out.println(nums[0]);
        }
    }
}
```

What does this code output ?

```java
public class Main {
    public static void main(String[] args) {
        String[] names = { "Sarah", "Jake", "Lee" };
        for (String name : names) {
            System.out.println(name);
        }
    }
}
```

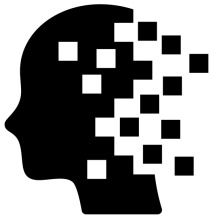
What does this code output ?

```java
public class Main {
    public static void main(String[] args) {
        int a = 2;
        double b = 3.0;
        int[] nums = {a, b};
        System.out.println(Arrays.toString(nums));
    }
}
```

What does this code output ?

```java
public class Main {
    public static void main(String[] args) {
        long[] numbers = new long[10];
        for (long number: numbers) {
            System.out.println(number);
        }
    }
}
```

Pause & Recall



Created by Victoruler
from Noun Project

Close your eyes and try to recall as many things as possible that were covered during this lesson.

Alternatively, you can keep your eyes open and write down as many things you remember on a piece of paper.

This will help strengthen your memory of key concepts 💪

Exercise 1 *(GenAI 🟠 Orange)*

Write a code that prints out all even numbers between 0 and 100.

How many different solutions can you find to solve this exercise ? Which one is your favorite solution and why ?

Exercise 2 *(GenAI 🟠 Orange)*

Construct a program that allows the user to enter a number. Output all numbers between 1 and that number.

Exercise 3 *(GenAI 🟠 Orange)*

Define an array with numbers

Write a Java program to sum values of that array.

Exercise 4 *(GenAI 🟠 Orange)*

Define an array with numbers

Write a Java program to calculate the average value of that array

Exercise 5 *(GenAI 🟠 Orange)*

Write a Java program to test if an array contains a specific value.

Exercise 6 *(GenAI 🟠 Orange)*

Write a Java program to find the index of an array element.

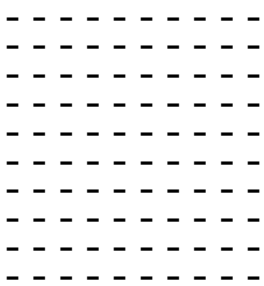
Exercise 7 *(GenAI 🟠 Orange)*

Write a Java program that takes an array of numbers and outputs all numbers below a certain threshold.

Exercise 8 *(GenAI 🟠 Orange)*

Write a Java program to print the following grid.

Expected Output :



Exercise 9 *(GenAI 🟠 Orange)*

Write a Java program to find the maximum and minimum value of an array.

Exercise 10 *(GenAI 🟠 Orange)*

In this exercise we are confronted with **parallel arrays**. Parallel arrays are a group of arrays of the same size where the element at a given index in one of the arrays corresponds to the element at the same index in another array.

Write the tracetable of the following code

```java
public class Main {
    public static void main(String[] args) {
        int[] grades = new int[] {4,6,3,6,7,5,7};
        String[] names = new String[]{"john","sarah","lilian","leo","max","lea","cecilia"};
        for (int i = 0; i < grades.length; i++) {
            if (grades[i] < 5) {
                System.out.println(names[i] + " should probably come to OH");
            }
        }
    }
}
```

Exercise 11 *(GenAI 🟠 Orange)*

Write the tracetable of the following code

```java
public class Main {
    public static void main(String[] args) {
        for (int i = 1; i <= 10; i++) {
            if (i % 3 == 0 && i % 5 ==0) {
                System.out.println("fizz-buzz");
            }
            else if (i % 3 == 0) {
                System.out.println("Fizz");
            }
            else if (i % 5 == 0) {
                System.out.println("Buzz");
            }
            else {
                System.out.println(i);
            }
        }
        System.out.println("The end");
    }
}
```


Exercise 12 *(GenAI 🟠 Orange)*

Write the tracetable of the following code

```java
public class Main {
    public static void main(String[] args) {
        int[] stock = new int[]{42,3,0,1,65,34,21,0,75,20};
        int count = 0;
        int total = 0;
        for (int i = 0; i < stock.length; i++) {
            if (stock[i] > 0) {
                count = count + 1;
                total = total + stock[i];
            }
        }

        if (count != 0){
            int average = total / count;
            System.out.println("Average stock is " + average);
        } else {
            System.out.println("All values are equal to 0");
        }
    }
```

Homework

Finish all the exercises

Create flashcards for the following terms:

1D array
Index
Parallel arrays
Sequence of instructions
Count controlled loop
How to write a trace table
Dry run testing
Static data structure
Direct access
Sequential access
Advantages and disadvantages of static data structures