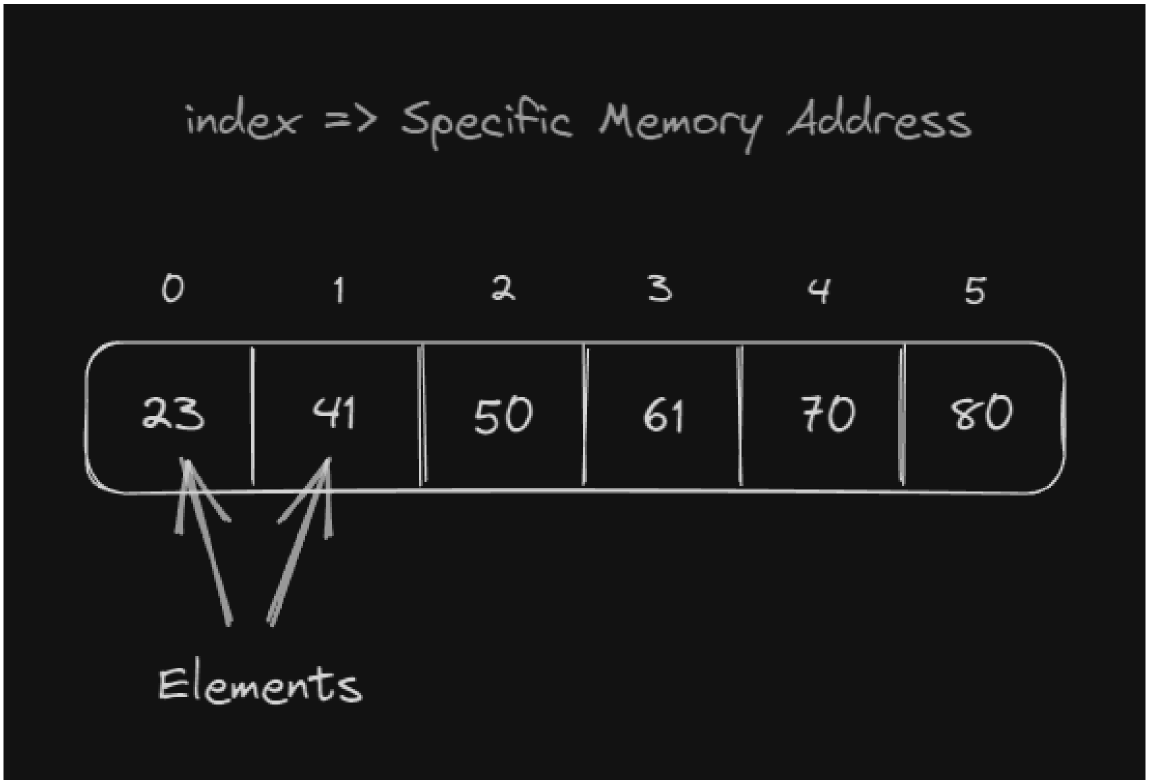
🗓️

21/09/2023

▼ Arrays

- An array is known as a specific data structure which is capable of storing similar type of values instead of declaring separate variables.
- When creating an array it should have a specific name which will considered as a `Unique Identifier` . When declaring an array, it should have a valid data type and array size is not mandatory.
- When accessing array elements we are using the array name along with the `array index`.
- Array indexes are always starting from 0 and it will be a unique memory address.



▼ Java Empty Array Declaration

- In Java there’s a possibility of declaring an array without values and without an array size.

Example

```
String[] cars;
```

- In above example [ ] defines cars is an array.

▼ Java Array with Fixed Size Declaration

- If you know the array size by using the following syntax a fixed size array can be declared.

Example

```
int[] intArray = new int[20];
```

Task 01

Create a Java program to get 10 integer inputs to an array and find the maximum input.

```
import java.util.Scanner;

public class findMax {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        int[] intArray = new int[10];
        int max = 0;

        for (int i=0; i<10; i++) {
            System.out.println("Enter a number: ");
            intArray[i] = input.nextInt();

            if (intArray[i] > max) {
                max = intArray[i];
            }
        }

        System.out.println("The max number is: " + max);
    }
}
```

▼ Java Classes

- In OOP the entire application can break down into different classes based on its functionality.
- Once the code is divided into classes, these classes can be reusable in the same project.

- In OOP, a class can contain `member variables` and `member functions` without having any limit.
- These functions and variables can be accessed by other classes using class objects.

> 💡 But in OOP accessibility will decide by `access specifiers`.
>
> There are 3 main access specifiers known as,
>
> 1. Public
> 2. Protected
> 3. Private

- If the variable or function is public that can be accessed by all other classes using a class object.

Example 01

```
class ClassNAme {
  // fields
  // methods
}
```

Example 02

```
class Bicycle {

  // state or field
  private int gear = 5;

  // behavior or method
  public void braking() {
    System.out.println("Working of Braking")
  }
}
```

Task 02

Write a Java main method to the above code and access the braking method.

```
import java.util.Scanner;

public class classes {
    public static void main(String[] args) {

        Bicycle objBicycle = new Bicycle();
        objBicycle.braking();
    }
}

class Bicycle {

    // state or field
    private int gear = 5;

    // behavior or method
    public void braking() {
        System.out.println(" Working of Braking");
    }
}
```