CJ Information

S – AP CS A

IC – Arrays in Java

HW - none

A – 2024.12.19 Thu – TEST(???)

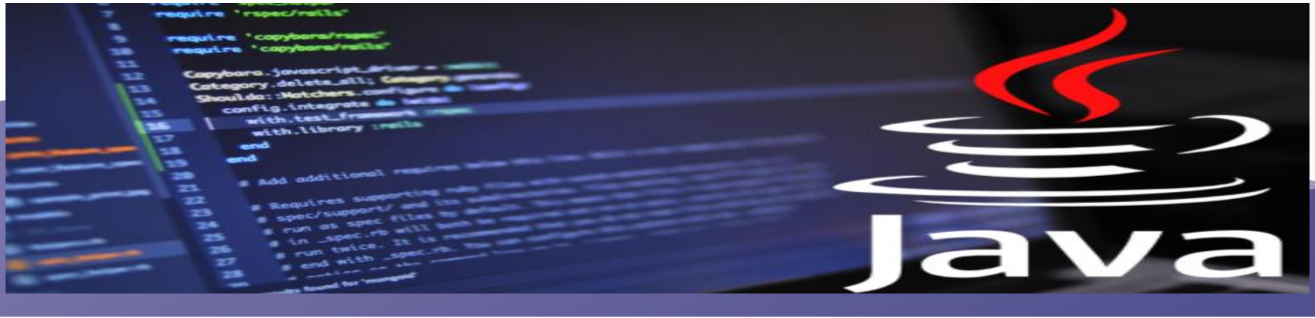
Arrays in Java

Outline :

- Arrays in general and in Java
- Arrays as a multiple container
- Example

What we learn today:

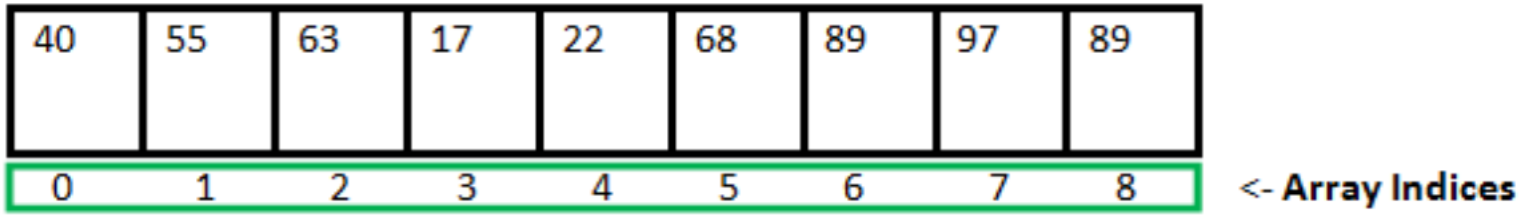
- What is an Array in Java
- How to store and retrieve information from an Array
- Examples



Arrays in Java

Creating and Access

An array is a block of memory that stores a collection of data items (elements) of the same type under one name. Arrays are useful whenever you have many elements of data of the same type that you want to keep track of, but you don't need to name each one. Instead you use the array name and a number (called an index) for the position of an item in the array. You can make arrays of ints, doubles, Strings, and even classes

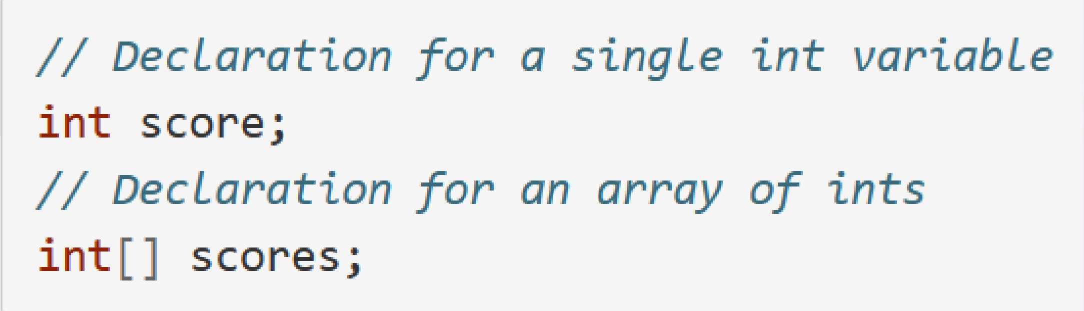


40	55	63	17	22	68	89	97	89	
0	1	2	3	4	5	6	7	8	<- Array Indices

Array Length = 9
First Index = 0
Last Index = 8

Arrays in Java

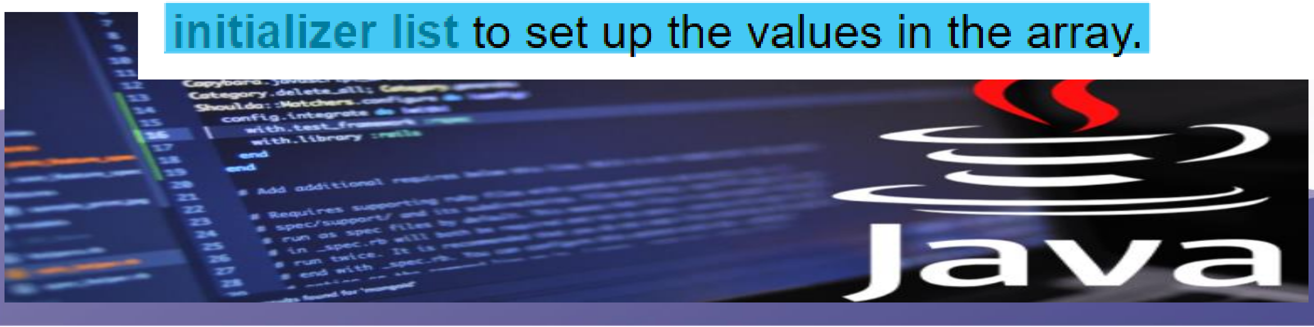
Declaring and Creating Array



```
// Declaration for a single int variable
int score;
// Declaration for an array of ints
int[] scores;
```

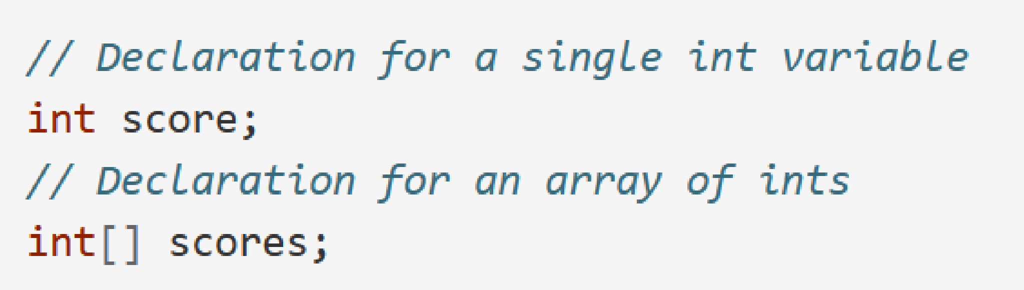
The declarations do not create the array. Arrays are **objects** in Java, so any variable that declares an array holds a reference to an object. If the array hasn't been created yet and you try to print the value of the variable, it will print **null** (meaning it doesn't reference any object yet).

There are two ways to create an array. You can use the keyword **new** to get new memory or use an **initializer list** to set up the values in the array.

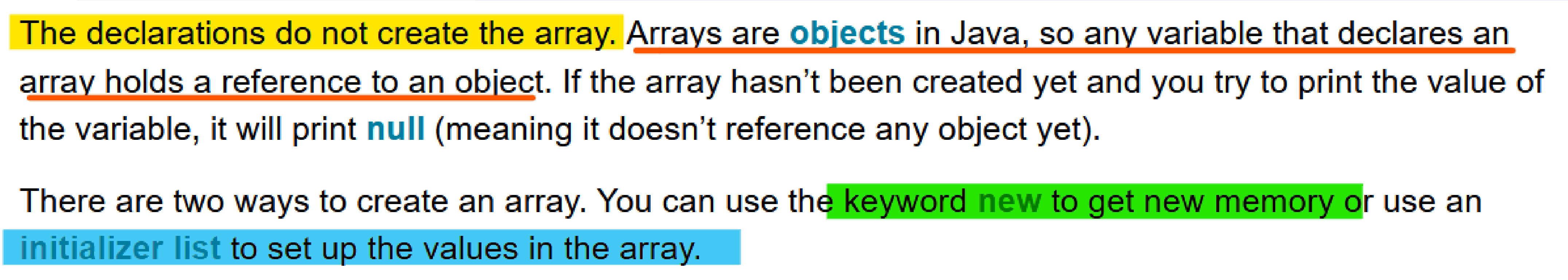


Arrays in Java

Declaring and Creating Array

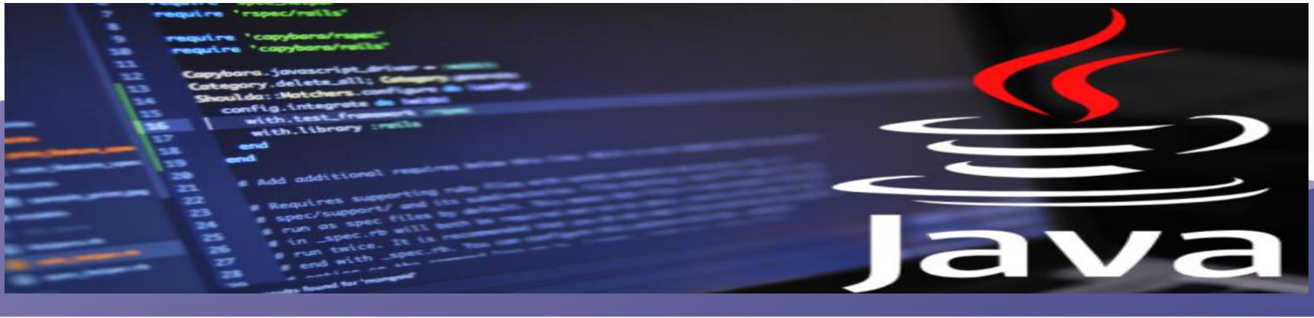


```
// Declaration for a single int variable
int score;
// Declaration for an array of ints
int[] scores;
```



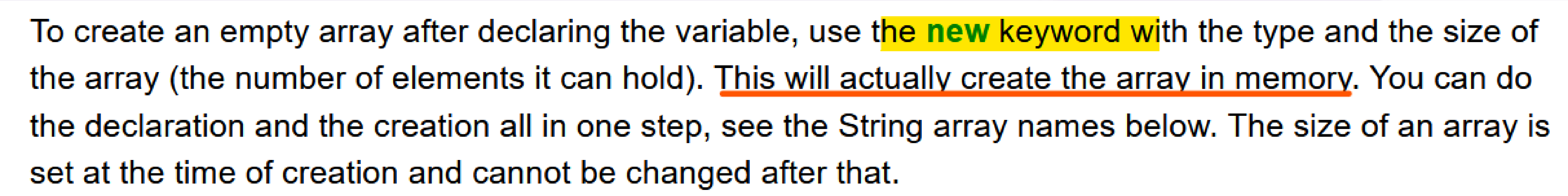
The declarations do not create the array. Arrays are **objects** in Java, so any variable that declares an array holds a reference to an object. If the array hasn't been created yet and you try to print the value of the variable, it will print **null** (meaning it doesn't reference any object yet).

There are two ways to create an array. You can use the keyword **new** to get new memory or use an **initializer list** to set up the values in the array.

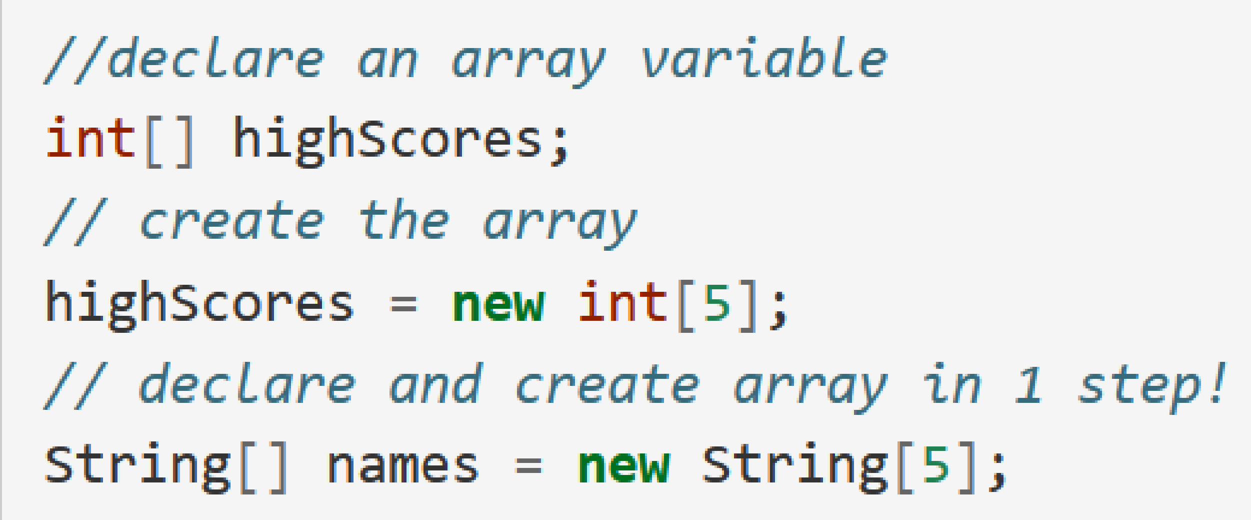


Arrays in Java

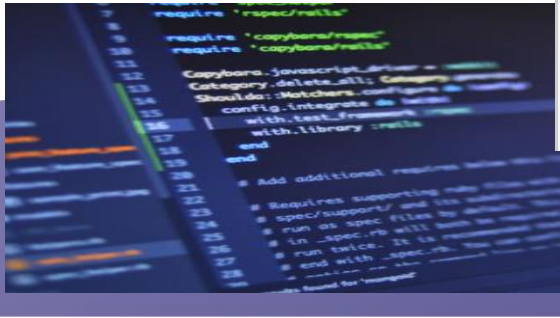
Using new to create an array



To create an empty array after declaring the variable, use the new keyword with the type and the size of the array (the number of elements it can hold). This will actually create the array in memory. You can do the declaration and the creation all in one step, see the String array names below. The size of an array is set at the time of creation and cannot be changed after that.



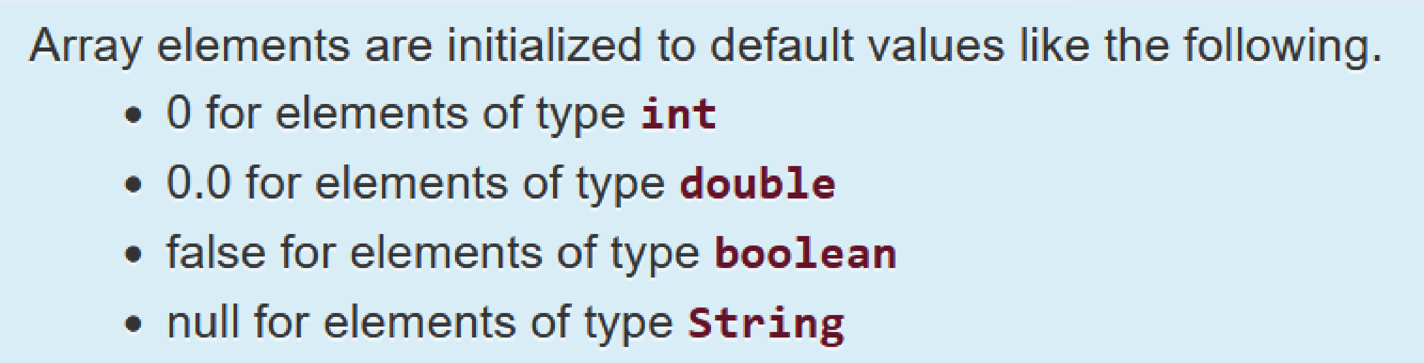
```
//declare an array variable
int[] highScores;
// create the array
highScores = new int[5];
// declare and create array in 1 step!
String[] names = new String[5];
```



Arrays in Java

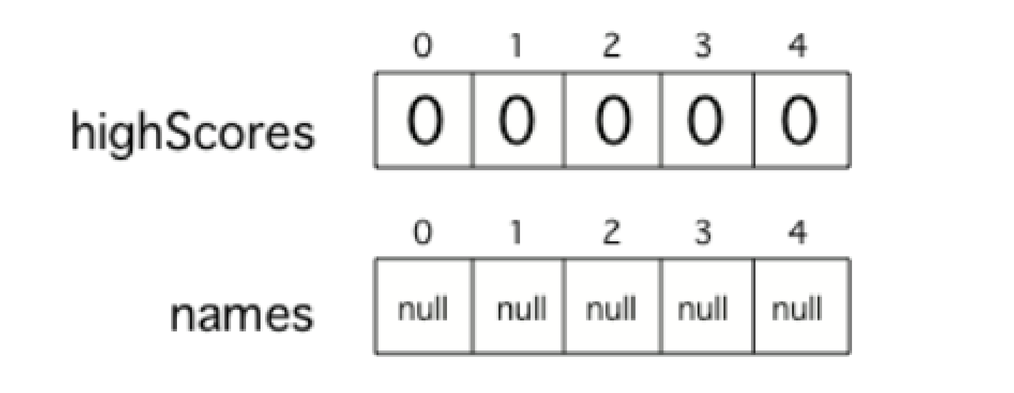
Using new to create an array

IMPORTANT

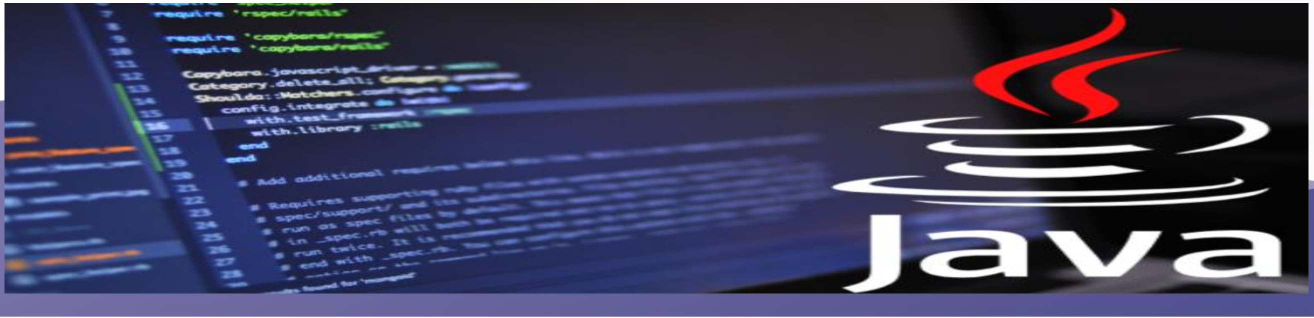


Array elements are initialized to default values like the following.

- 0 for elements of type **`int`**
- 0.0 for elements of type **`double`**
- false for elements of type **`boolean`**
- null for elements of type **`String`**

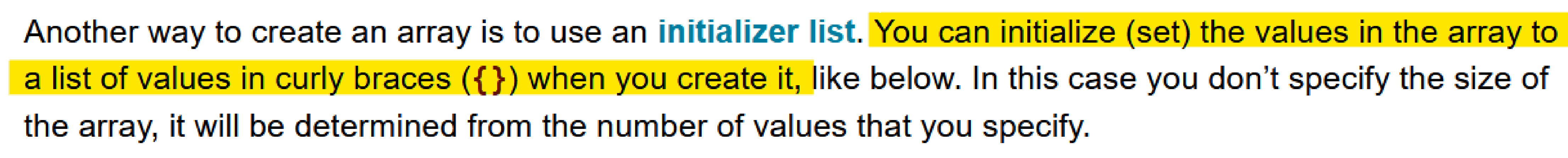


	0	1	2	3	4
highScores	0	0	0	0	0
	0	1	2	3	4
names	null	null	null	null	null

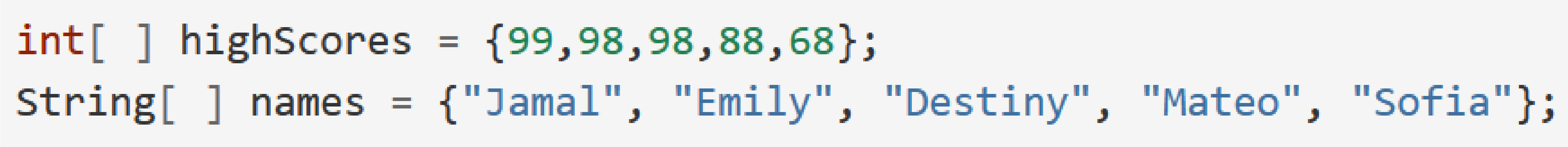


Arrays in Java

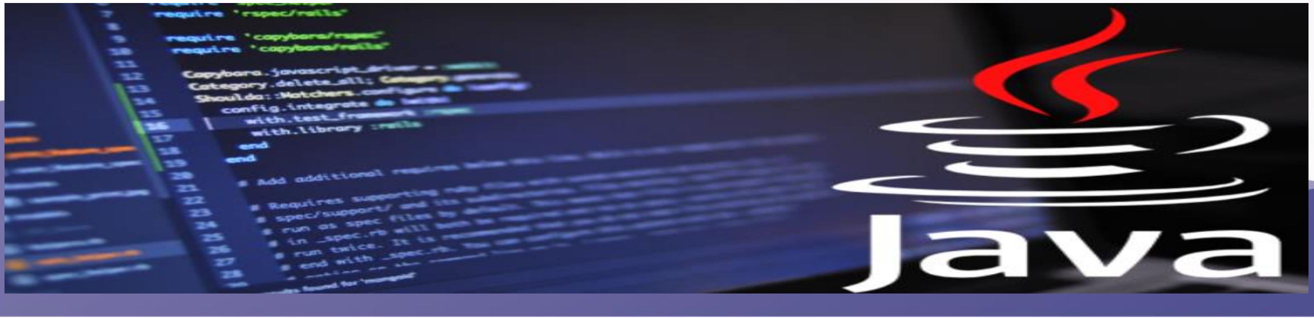
Using initializer list to create an array



Another way to create an array is to use an **initializer list**. You can initialize (set) the values in the array to a list of values in curly braces (`{}`) when you create it, like below. In this case you don’t specify the size of the array, it will be determined from the number of values that you specify.

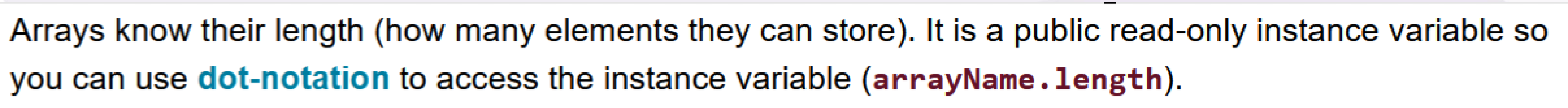


```
int[ ] highScores = {99,98,98,88,68};
String[ ] names = {"Jamal", "Emily", "Destiny", "Mateo", "Sofia"};
```



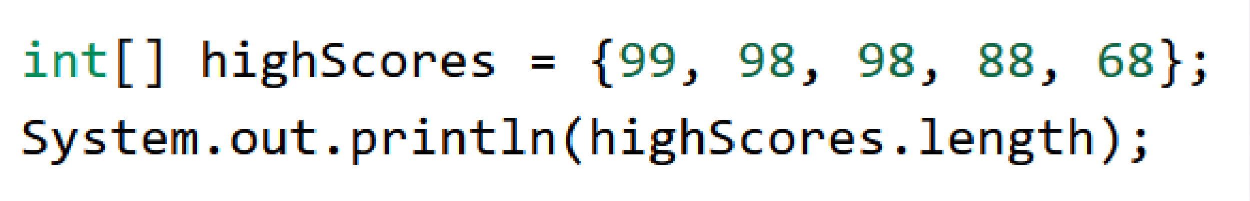
Arrays in Java

Using array length



Arrays know their length (how many elements they can store). It is a public read-only instance variable so you can use **dot-notation** to access the instance variable (`arrayName.length`).

Example



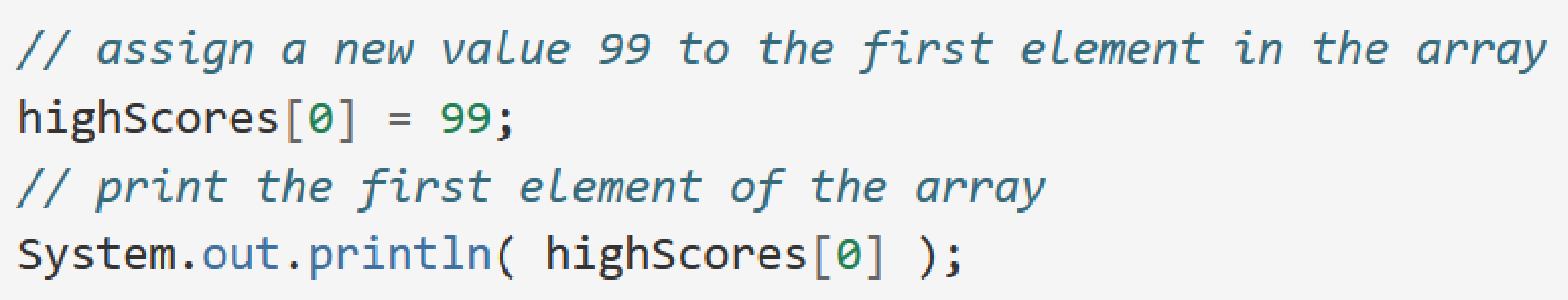
```
int[] highScores = {99, 98, 98, 88, 68};
System.out.println(highScores.length);
```

What is the result ???

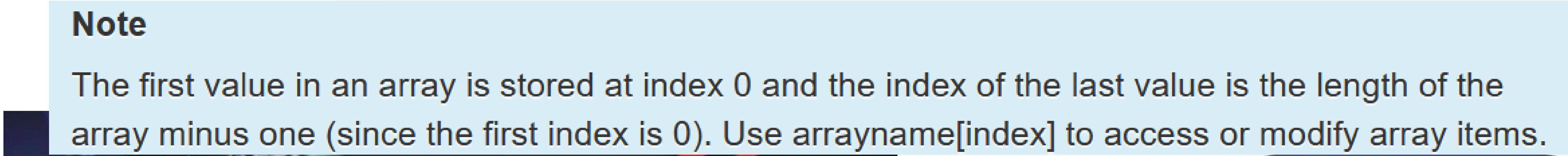
Arrays in Java

Access and Modify Array Values

To access an element of the array, we use an **index**, a variable that identify a specific position inside the array.

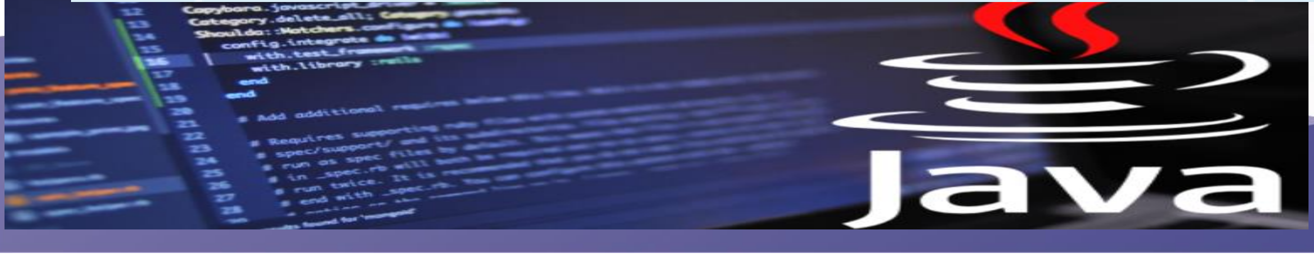


```
// assign a new value 99 to the first element in the array
highScores[0] = 99;
// print the first element of the array
System.out.println( highScores[0] );
```



Note

The first value in an array is stored at index 0 and the index of the last value is the length of the array minus one (since the first index is 0). Use arrayname[index] to access or modify array items.



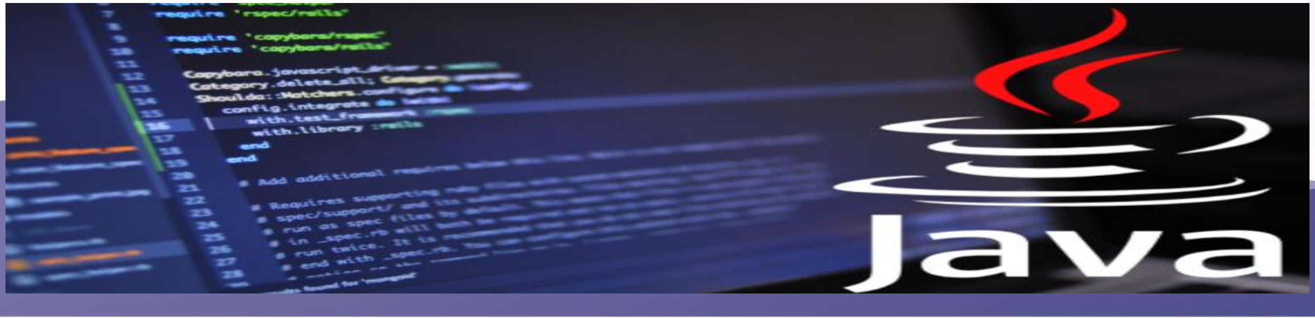
Arrays in Java

Access and Modify Array Values

WARNING !!!!!!

Q : What happens if you try to access an element at position that is not present in the array????

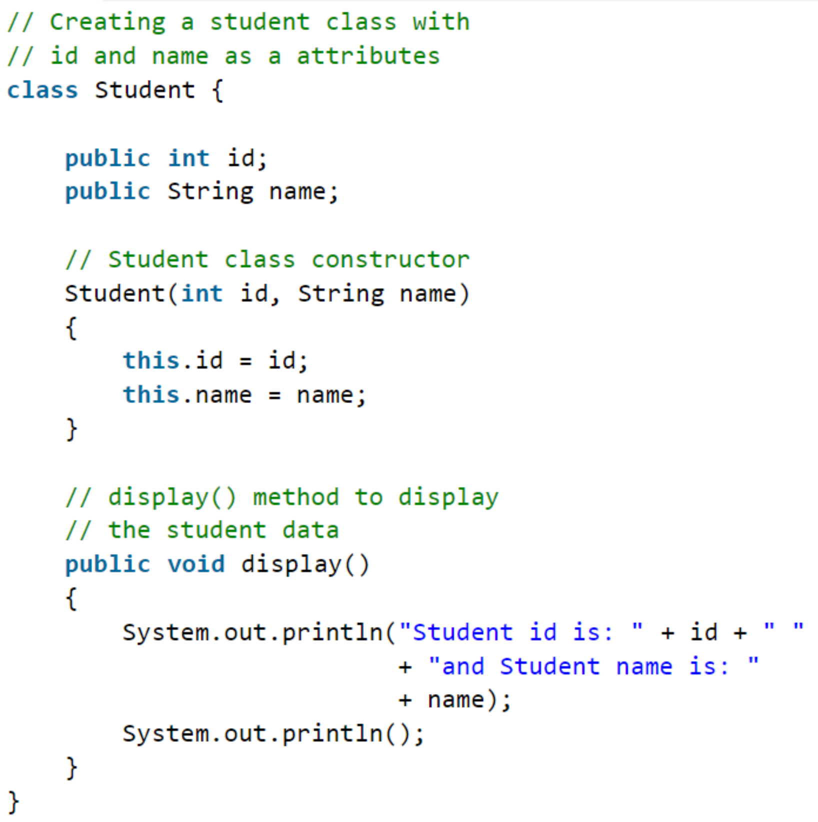
A : you will get an error message called `ArrayIndexOutOfBoundsException.`
and your program stops working !!!!!!!!!!!!



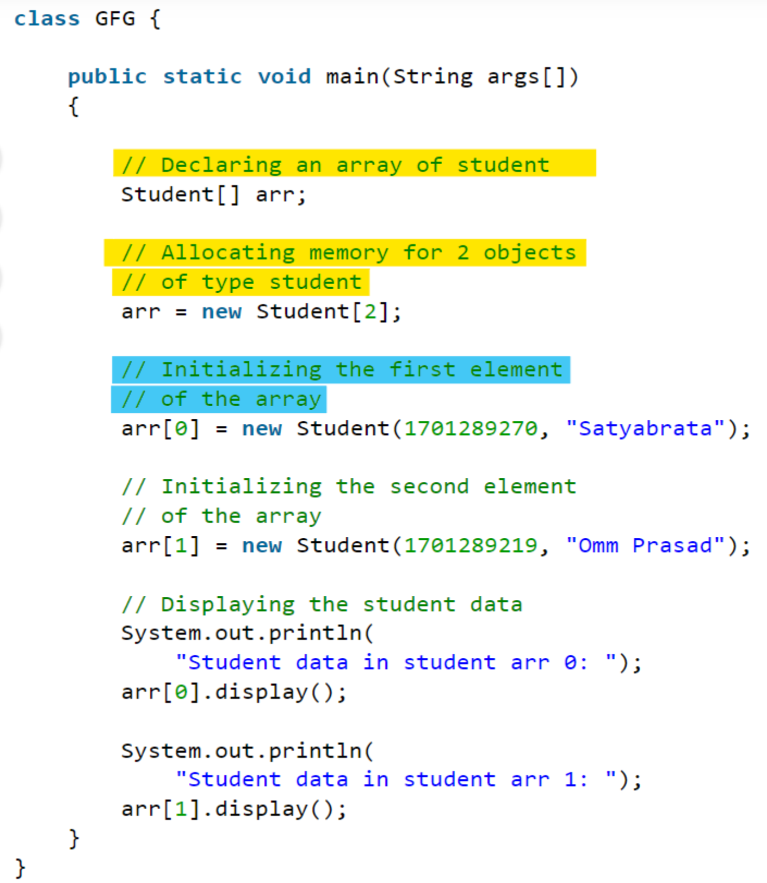
Arrays in Java

Array of Objects.

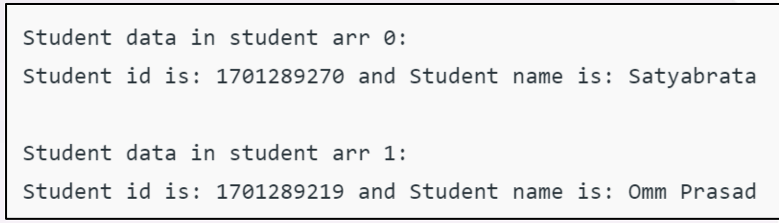
Array can contain also complex data structures as objects. Have a look at the following example to better understand how it works:



```
// Creating a student class with
// id and name as a attributes
class Student {

    public int id;
    public String name;

    // Student class constructor
    Student(int id, String name)
    {
        this.id = id;
        this.name = name;
    }

    // display() method to display
    // the student data
    public void display()
    {
        System.out.println("Student id is: " + id + " "
                           + "and Student name is: "
                           + name);
        System.out.println();
    }
}
```



```
class GFG {

    public static void main(String args[])
    {

        // Declaring an array of student
        Student[] arr;

        // Allocating memory for 2 objects
        // of type student
        arr = new Student[2];

        // Initializing the first element
        // of the array
        arr[0] = new Student(1701289270, "Satyabrata");

        // Initializing the second element
        // of the array
        arr[1] = new Student(1701289219, "Omm Prasad");

        // Displaying the student data
        System.out.println(
            "Student data in student arr 0: ");
        arr[0].display();

        System.out.println(
            "Student data in student arr 1: ");
        arr[1].display();
    }
}
```

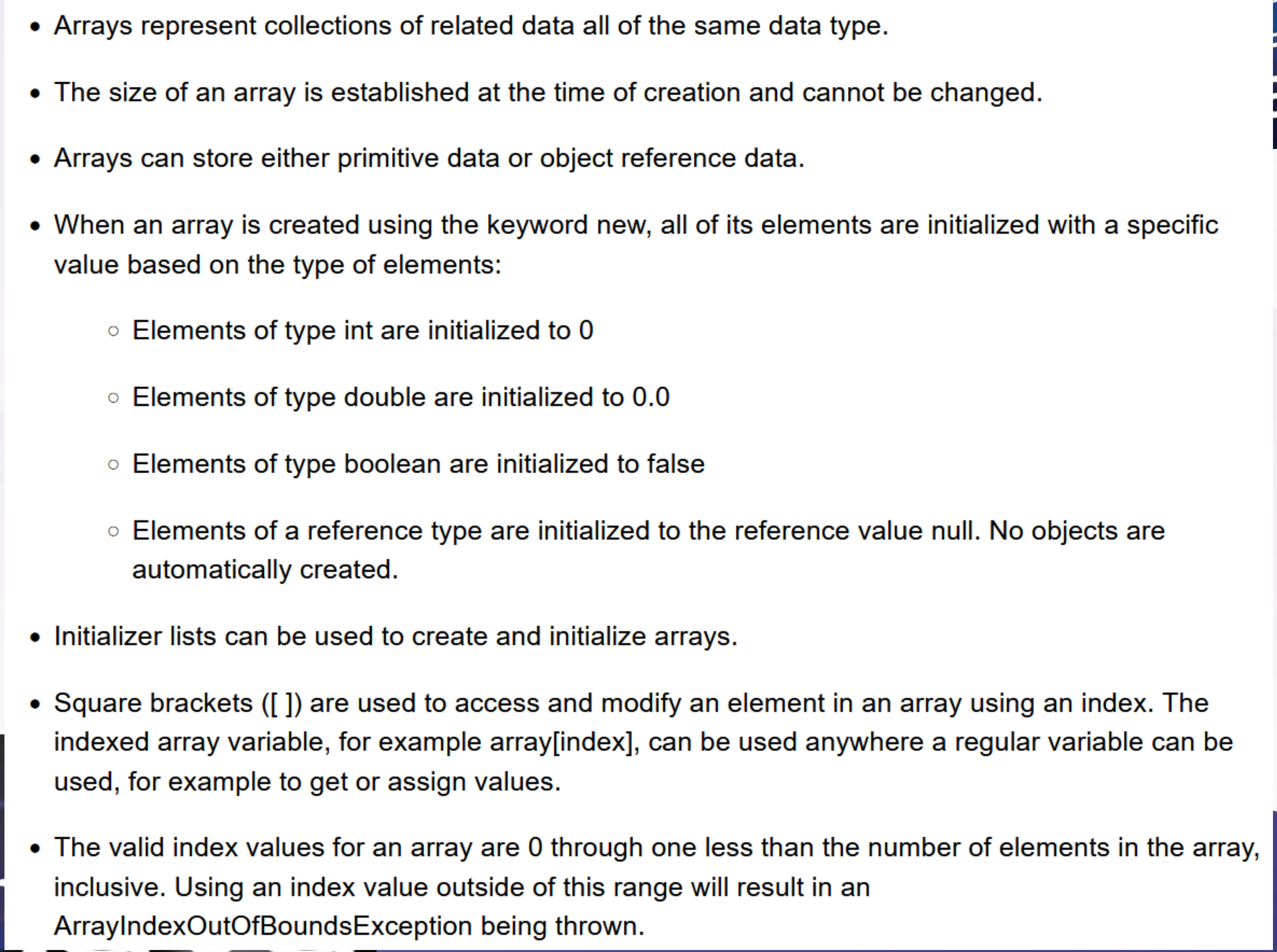


```
Student data in student arr 0:
Student id is: 1701289270 and Student name is: Satyabrata

Student data in student arr 1:
Student id is: 1701289219 and Student name is: Omm Prasad
```

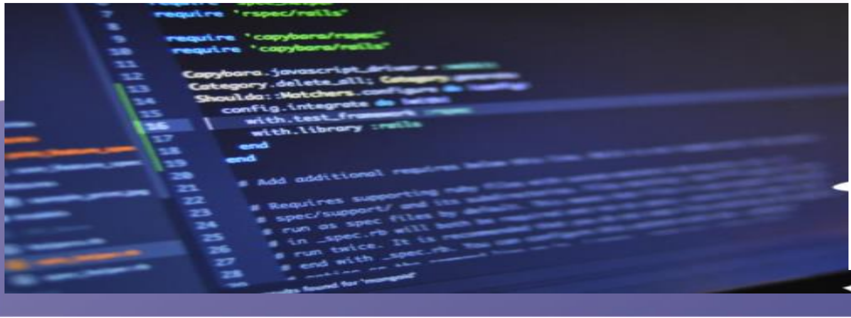
Arrays in Java

Access and Modify Array Values

SUMMARY



- Arrays represent collections of related data all of the same data type.
- The size of an array is established at the time of creation and cannot be changed.
- Arrays can store either primitive data or object reference data.
- When an array is created using the keyword new, all of its elements are initialized with a specific value based on the type of elements:
  - Elements of type int are initialized to 0
  - Elements of type double are initialized to 0.0
  - Elements of type boolean are initialized to false
  - Elements of a reference type are initialized to the reference value null. No objects are automatically created.
- Initializer lists can be used to create and initialize arrays.
- Square brackets ([ ]) are used to access and modify an element in an array using an index. The indexed array variable, for example array[index], can be used anywhere a regular variable can be used, for example to get or assign values.
- The valid index values for an array are 0 through one less than the number of elements in the array, inclusive. Using an index value outside of this range will result in an ArrayIndexOutOfBoundsException being thrown.



Arrays in Java

Access and Modify Array Values

Exercise

Create a Java program that implements the following points:

1) Create a Car class with the following ATTRIBUTES : **power** (of type int), **color** (of type string) , **fuel** (of type string) and the **toString**() method that prints out the information about the car

2) Create a TestingCar class containing an array of 2 objects of type Car.
3) Initialize the two objects and insert them inside the array
4) Using a for loop, you print out the information for every element of the array.