COMP90041

Programming and Software Development

Semester 1, 2021

Lab 6

Andrew Naughton

andrew.naughton@unimelb.edu.au

THE UNIVERSITY OF MELBOURNE

Outline

- Arrays
- Creating an array (2 options)
- Array Properties
- Looping through an Array (2 options)
- Deep copy of Arrays
- Use of == with Arrays
- Exercises

THE UNIVERSITY OF MELBOURNE

Problem context

- Say we want to store the grades of 50 students – how do we do it?
- Can we loop through these 50 variables easily?
- What if we wish to pass these grades to a function?
  - That's a lot of parameters
- Is there a way to do it with one variable?
  - …Enter the array!

```
int s1_grade;
int s2_grade;
int s3_grade;
int s4_grade;
      .
      .
      .
int s50_grade;
```

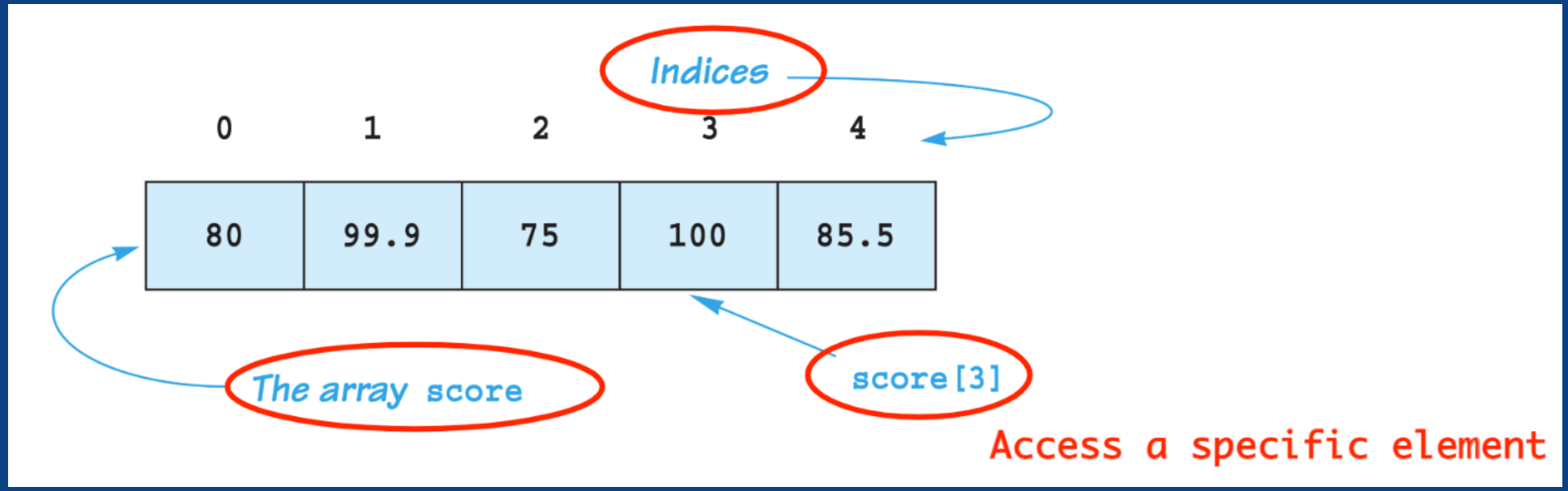
THE UNIVERSITY OF MELBOURNE

Arrays

- A basic data structure
- Store multiple values in a single variable, instead of declaring separate variables for each value

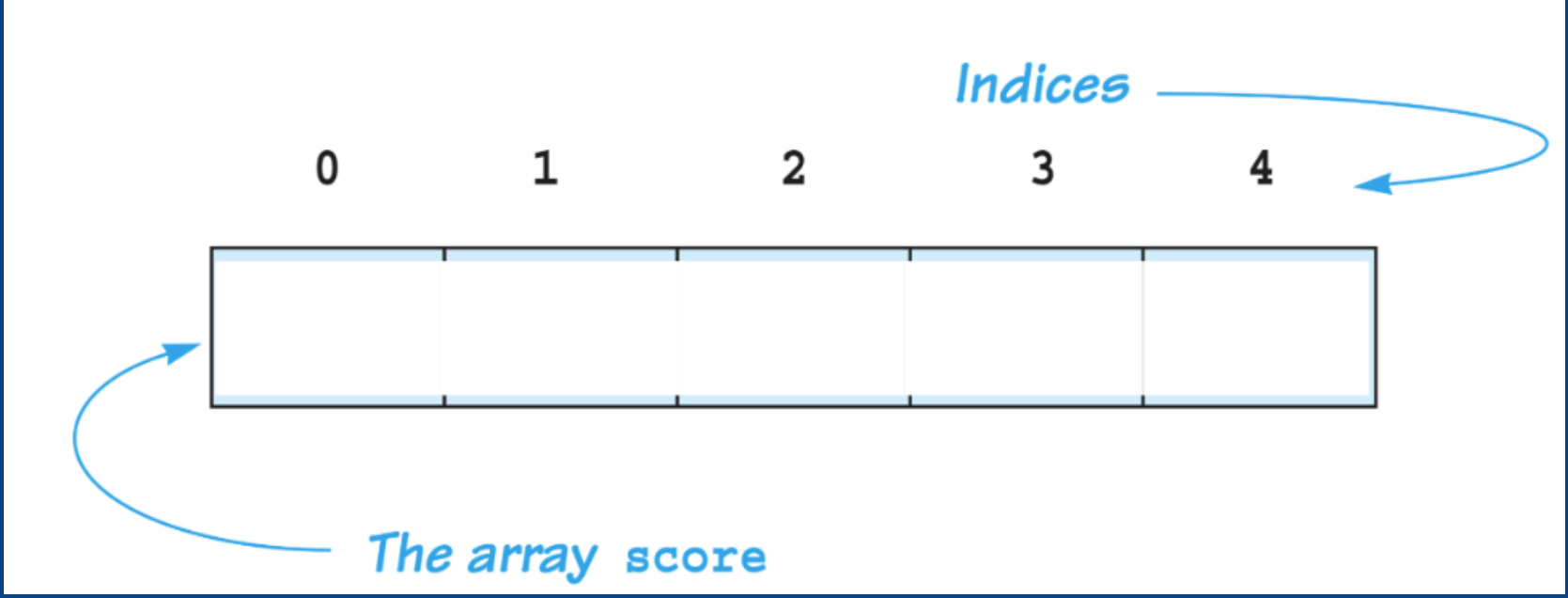
Creating an Array

- Two ways to create an array:
- Option 1
  - `double[] score = {80, 99.9, 75, 100, 85.5};`



Creating an Array

- Two options to create an array:
- Option 2
  - `double[] score = new double[5];`



THE UNIVERSITY OF MELBOURNE

Array Properties

- `double[] score = {80, 99.9, 75, 100, 85.5};`
- `double[] score = new double[5];`

- Size of the array has to be predefined
- Fixed size (static array)
- Stores one data type (primitive or Class Type)

Array Properties

- `double[] score = {80, 99.9, 75, 100, 85.5}`
- `double[] score = new double[5];`

- How can we change array elements?
  - `score[2] = 77;`
- How can we check the size of array?
  - `score.length`
- What if we try access outside the array length?
  - `System.out.println(score[5]);`
  - `// note: the last index is length - 1`
  - JVM throws `ArrayIndexOutOfBoundsException`

THE UNIVERSITY OF MELBOURNE

Looping through an Array

- Two options to loop through an array:

```
String[] cars = {"Volvo", "BMW", "Ford"};
```

- Option 1: Loop over indices

```
for (int i = 0; i < cars.length; i++) {
  System.out.println(cars[i]);
}
```

- Option 2: For each loop

```
for (String car : cars) {
    System.out.println(car);
}
```

THE UNIVERSITY OF MELBOURNE

Deep copy of Arrays

- Means the copied array is a separate object with same contents
- For primitives:

```
int[] intArray = {1,2,3};
int[] cloneArray = intArray.clone();
```

- For Class Base Types:
  - Copy constructor (seen last week)
  - `Date copiedDate = new Data(originalDate);`
  - `// tries to call copy constructor of Date`

THE UNIVERSITY OF MELBOURNE

Use of == with Arrays

- `==` Only tests if two arrays are stored in the same location in the computer's memory
  - I.e. it does not test if they contain the same values

- Instead, use `Arrays.equals(intArray, cloneArray);`
- For Class type, must `@Override` the Class' `equals` method

- False
- True

```
System.out.println(cloneArray == intArray);
System.out.println(Arrays.equals(cloneArray, intArray));
```

THE UNIVERSITY OF
MELBOURNE

Exercises