Arrays and ArrayLists

Little boxes, on a hillside, little boxes made of ticky-tacky
Little boxes, little boxes, little boxes, all the same
There’s a green one and a pink one and a blue one and a yellow one
And they're all made out of ticky-tacky and they all look just the same

Note: These slides are the ones that were used in the lecture. They are substantially different from the textbook’s slides.

CS101 @ Özyeğin University

Slides are adapted from the originals available at http://www-cs-faculty.stanford.edu/~eroberts/books/ArtAndScienceOfJava/
Motivating Example

• Get a set of numbers from the user
• Print out the numbers that are above the average of the whole set of numbers
Motivating Example

- Get a set of numbers from the user
- Print out the numbers that are above the average of the whole set of numbers
- Requires us to process the set of numbers twice
  - First, to calculate the average
  - Second, to pick up numbers that are above the average
- Hence, requires us to store these numbers somehow
The Need for a Data Structure

- We cannot ask the user to enter the numbers again
  - Inconvenient
  - Error-prone

- We cannot store numbers in separate variables
  - Not practical for large amount of numbers
  - Total amount of numbers is not always known priori
Definition: Arrays

A named, ordered collection of variables of identical type

• We do not name every individual element in the collection
• Instead, we give a name to the whole collection
• We access individual elements in the collection by specifying its order in the collection
  − This is called the index
Introduction to Arrays

• An array is a collection of individual data values
• An array is
  1. ordered. You can say things like “here is the first”, “here is the second”, and so on.
  2. homogeneous. Every value in the array must have the same type. An integer array contains integers only.
• The number of elements is called the length of the array.
• Each element is identified by its position number in the array, which is called its index.
• In Java, index numbers always begin with 0 and therefore extends up to one less than the length of the array.
An Example of Array Declaration

- This easiest way to visualize arrays is to think of them as a sequence of boxes.

```java
int score = 0;
int[] scores = new int[10];
```

- Java automatically initializes each element of a newly created array to its default value, which is zero for numeric types, false for values of type boolean, and null for objects.
Array Selection

- Use the array selection syntax to access and change the elements of an array.

```java
int[] scores = new int[10];

scores

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

scores[0] = 25;

scores

<table>
<thead>
<tr>
<th>25</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

scores[1] = 17;

scores

<table>
<thead>
<tr>
<th>25</th>
<th>17</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>
Array Selection

- Use the array selection syntax to access and change the elements of an array.

```java
int[] scores = new int[10];
println("First element is " + scores[0]);
println("Second element is " + scores[1]);
println("Third element is " + scores[2]);
println("---------------------");
scores[0] = 25;
scores[1] = 17;
println("First element is " + scores[0]);
println("Second element is " + scores[1]);
println("Third element is " + scores[2]);
println("---------------------");
scores[2] = scores[0] + scores[1];
println("First element is " + scores[0]);
println("Second element is " + scores[1]);
println("Third element is " + scores[2]);
println("---------------------");
```

First element is 0
Second element is 0
Third element is 0
---------------------
First element is 25
Second element is 17
Third element is 0
---------------------
First element is 25
Second element is 17
Third element is 42
---------------------
Array Length

- You can query the length of an array by accessing its `length` field.

```java
int[] scores = new int[10];
println("Length of the array is "+ scores.length);
```

Length of the array is 10
Exercise 1

• What does the following program print?

```java
int[] scores = new int[10];
scores[5] = scores.length;
println(scores[5]);
```

A) 42
B) 11
C) 10
D) There is an error
Exercise 2

• What does the following program print?

```java
int[] scores = new int[10];
scores[5] = scores.length;
scores[5]++;
scores[1]++;
println(scores[5] + scores[1]);
```

A) 10
B) 11
C) 12
D) There is an error
Exercise 3
• What does the following program print?

```java
int[] scores = new int[10];
scores[scores.length] = 42;
println(scores[10]);
```

A) 10
B) 11
C) 42
D) There is an error

No element at index 10!
Exercise 3

• What does the following program print?

```java
int[] scores = new int[10];
scores[scores.length] = 42;
println(scores[10]);
```

```
java.lang.ArrayIndexOutOfBoundsException: 10
    at ch11.ArrayElementAccess.run(ArrayElementAccess.java:13)
    at acm.program.Program.runHook(Program.java)
    at acm.program.Program.startRun(Program.java)
    at acm.program.Program.start(Program.java)
    at sun.applet.AppletPanel.run(AppletPanel.java:464)
    at java.lang.Thread.run(Thread.java:680)
```
Exercise 4

- How does the array look like after executing

```java
int[] scores = new int[10];
for(int i=0; i < scores.length; i++) {
    scores[i] = i;
}
```

A) 0 1 2 3 4 5 6 7 8 9
B) 0 0 0 0 0 0 0 0 0 0
C) 10 10 10 10 10 10 10 10 10 10
D) 1 2 3 4 5 6 7 8 9 10
Exercise 5

Write a method `sumArray` that takes an array of integers and returns the sum of those values.

```java
public int sumArray(int[] array) {
    int sum = 0;
    for (int i = 0; i < array.length; i++) {
        sum += array[i];
    }
    return sum;
}
```

D) All
Exercise (printing an array)

Write a program that prints an array of integers as a comma-separated string.

```java
int[] squares = new int[10];
for(int i=0; i < squares.length; i++) {
    squares[i] = i*i;
}

print("[");
for(int i=0; i < squares.length; i++) {
    if(i > 0) {
        print(",");
    }
    print(squares[i]);
}
println("]");
```

[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
Arrays in Java are implemented as objects, which means that they are stored in the heap. The value stored in an array variable is simply a reference to the actual array.

```java
int score = 0;
int[] scores = new int[10];
```
Internal Representation of Arrays

```java
int score = 5;
int foo = score;
foo++; // prints 5
println(score); // prints 5
println(foo);  // prints 6
```
Internal Representation of Arrays

```java
int score = 5;
int foo = score;
foo++;
println(score);  // prints 5
println(foo);    // prints 6
```
int score = 5;
int foo = score;
foo++;
println(score); // prints 5
println(foo);   // prints 6

<table>
<thead>
<tr>
<th>score</th>
<th>foo</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>
Internal Representation of Arrays

```java
int score = 5;
int foo = score;
foo++;
println(score); // prints 5
println(foo);   // prints 6
```

<table>
<thead>
<tr>
<th></th>
<th>score</th>
<th>foo</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>
Internal Representation of Arrays

```java
int[] scores = new int[10];
int[] numbers = scores;
numbers[5] = 42;
println(scores[5]);  // prints 42
println(numbers[5]);  // prints 42
scores[0] = 77;
println(scores[0]);  // prints 77
println(numbers[0]);  // prints 77
```
Internal Representation of Arrays

int[] scores = new int[10];
int[] numbers = scores;
numbers[5] = 42;
println(scores[5]);   // prints 42
println(numbers[5]);   // prints 42
scores[0] = 77;
println(scores[0]);   // prints 77
println(numbers[0]);   // prints 77
Internal Representation of Arrays

```java
int[] scores = new int[10];
int[] numbers = scores;
numbers[5] = 42;
println(scores[5]); // prints 42
println(numbers[5]); // prints 42
scores[0] = 77;
println(scores[0]); // prints 77
println(numbers[0]); // prints 77
```
Internal Representation of Arrays

```java
int[] scores = new int[10];
int[] numbers = scores;
numbers[5] = 42;
println(scores[5]);   // prints 42
println(numbers[5]);  // prints 42
scores[0] = 77;
println(scores[0]);   // prints 77
println(numbers[0]);  // prints 77
```
Internal Representation of Arrays

```java
int[] scores = new int[10];
int[] numbers = scores;
numbers[5] = 42;
println(scores[5]);    // prints 42
println(numbers[5]);   // prints 42
scores[0] = 77;
println(scores[0]);    // prints 77
println(numbers[0]);   // prints 77
```
Internal Representation of Arrays

```java
int[] scores = new int[10];
int[] numbers = scores;
numbers[5] = 42;
println(scores[5]);  // prints 42
println(numbers[5]);  // prints 42
scores[0] = 77;
println(scores[0]);  // prints 77
println(numbers[0]);  // prints 77
```
Exercise (Swapping)

• Write a program that swaps the values of two integer variables.

```java
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    first = second;
    second = first;

    println("First value after swapping: " + first);
    println("Second value after swapping: " + second);
}
```
Exercise (Swapping)

• What’s printed by the program below, if `first` is 17, `second` is 25?

```java
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    first = second;
    second = first;

    println("First value after swapping: " + first);
    println("Second value after swapping: " + second);
}
```

A) 17, 25
B) 25, 17
C) 17, 17
D) 25, 25
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    first = second;
    second = first;

    println("First value after swapping: " + first);
    println("Second value after swapping: " + second);
}
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    first = second;
    second = first;

    println("First value after swapping: " + first);
    println("Second value after swapping: " + second);
}
Exercise (Swapping)

```java
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    first = second;
    second = first;

    println("First value after swapping: " + first);
    println("Second value after swapping: " + second);
}
```

```
25
first
25
second
```

```
Enter 1st: 17
Enter 2nd: 25
```
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    first = second;
    second = first;

    println("First value after swapping: " + first);
    println("Second value after swapping: " + second);
}
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    first = second;
    second = first;

    println("First value after swapping: " + first);
    println("Second value after swapping: " + second);
}
Exercise (Swapping)

• What’s printed by the program below, if `first` is 17, `second` is 25?

```java
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    first = second;
    second = first;

    println("First value after swapping: " + first);
    println("Second value after swapping: " + second);
}
```

A) 17, 25  C) 17, 17  B) 25, 17  D) 25, 25
Exercise (Swapping)

• Write a program that swaps the values of two integer variables.

Correct version:

```java
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    int temp = first;
    first = second;
    second = temp;

    println("First value after swapping: " + first);
    println("Second value after swapping: " + second);
}
```
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    int temp = first;
    first = second;
    second = temp;

    println("First value after swapping: "+ first);
    println("Second value after swapping: "+ second);
}
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    int temp = first;
    first = second;
    second = temp;

    println("First value after swapping: " + first);
    println("Second value after swapping: " + second);
}
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    int temp = first;
    first = second;
    second = temp;

    println("First value after swapping: "+ first);
    println("Second value after swapping: "+ second);
}
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    int temp = first;
    first = second;
    second = temp;

    println("First value after swapping: " + first);
    println("Second value after swapping: " + second);
}
public void run() {
    int first = readInt("Enter 1st: ");
    int second = readInt("Enter 2nd: ");

    // swap the values
    int temp = first;
    first = second;
    second = temp;

    println("First value after swapping: "+ first);
    println("Second value after swapping: "+ second);
}
Exercise: Swapping Array Elements

- Write a program that swaps the first and last values of an array.

```java
int[] squares = new int[10];
for(int i=0; i < squares.length; i++) {
    squares[i] = i*i;
}
println(squares[0]); // prints 0
println(squares[squares.length - 1]); // prints 81
// write your code here
```
Swapping Array Elements

• Write a program that swaps the first and last values of an array.

```java
int[] squares = new int[10];
for (int i=0; i < squares.length; i++) {
    squares[i] = i*i;
}
println(squares[0]); // prints 0
println(squares[squares.length - 1]); // prints 81

// swap elements
int temp = squares[0];
squares[0] = squares[squares.length-1];
squares[squares.length-1] = temp;

println(squares[0]); // prints 81
println(squares[squares.length - 1]); // prints 0
```
Swapping Array Elements

• Write a Java method named `swapElements` that takes an integer array and two indices, and swaps the elements.

```java
public void swapElements(int[] array, int i, int j) {
    int temp = array[i];
    array[i] = array[j];
    array[j] = temp;
}
```