COMPSCI 105

Principles of Computer Science

Introduction
Nested Loops
Multi-Dimensional Arrays

Teaching staff

Andrew Luxton-Reilly

- Rm: 303.479
- Ext: 85654
- Email: andrew@cs.auckland.ac.nz

Robert Sheehan

- Rm: 303.371
- Ext: 88814
- Email: robert-s@cs.auckland.ac.nz

Course Ground Rules

Behaviour

- Respect other people
- One person speaks at a time
- Questions welcome any time
- Answers expected when questions asked

Active learning

- Expect you to think
- Expect you to work on exercises

Expect you to keep up with the class

- Read Internet resources regularly
- Find out the answers to questions

Independent Learning

- Experimentation

Course Content

Java programming constructs

- Nested loops
- Multi-dimensional arrays
- Exceptions and exception handling
- Text I/O

Big Ideas

- Recursion
- Measuring efficiency
  - Sorting algorithms
- Abstract Data Types
  - Linked Lists
  - Stacks and Queues
  - Trees
  - Priority queues, heaps
  - Hash tables

Course Requirements

Required reading

- Data Abstraction and Problem Solving with Java (Walls and Mirrors)
  - Frank Carrano
  - Janet Prichard

Assessment

- Assignments: 25%
  - Assignment 1: 5%,
  - Assignment 2: 10%,
  - Assignment 3: 10%,
- Test: 10%
- Exam: 65%
- Must pass both practical and theory

Tutorials

- Not Assessed

Important Dates

Assignment One due date:

Assignment Two due date:

Assignment Three due date:

Test date:

Exam date:

What can you remember from 101?

Style issues

- Naming conventions
- Indentation conventions

Syntax issues

- Variables, Constants and Literals
- Primitive Data Types and Object types
- Expressions and Operators
- Arrays
- Selection Statements
- Iteration Statements
- Classes and Methods

Techniques

- Tracing code
- Debugging
- Functional Decomposition

Exercises

(i) Explain what the following code does.

```
int m = a[0];
for (int i = 0; i < a.length; i++) {
       if (a[i] > m) {
              m = a[i];
       }
}
```

(ii) Explain what the following code does.

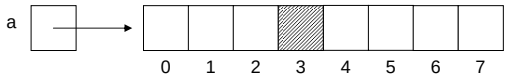
```
int m = a[0];
for (int i = 0; i < a.length; i++) {
       if (a[m] <= a[i]) {
              m = i;
       }
}
```

One-dimension arrays

Declare: `int[] a`

Use: `a[3]`

Visualise:

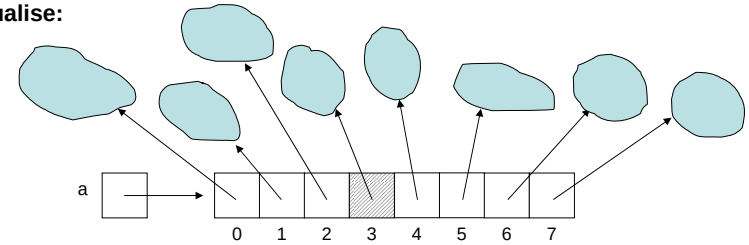


Arrays of objects

Declare: `String[] a`

Use: `a[3]`

Visualise:



Discussion questions

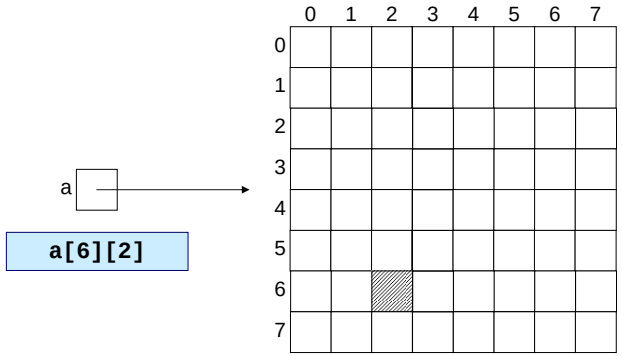
Could we have an array of arrays?

Could the arrays be different lengths?

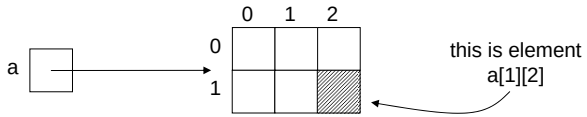
What would the syntax look like?

How could be visualise such a thing?

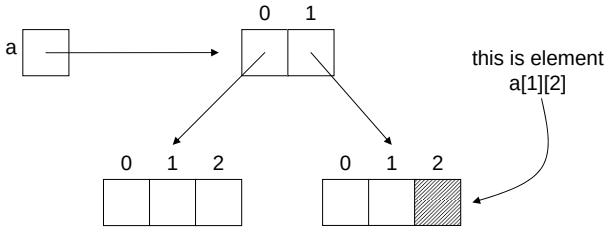
Two-dimensional arrays



The real picture?



A two-dimensional array is really just a one-dimensional array of one-dimensional arrays



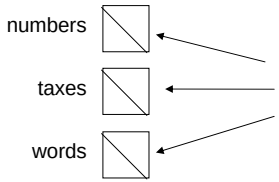
Syntax of two-dimensional arrays

Declaring an array

- Type, followed by two sets of square brackets [][]

```
int[][] numbers;    // a 2-d array of ints
double[][] taxes;   // a 2-d array of doubles
String[][] words;   // a 2-d array of Strings
```

Declaring only sets aside memory for the variable, not the information

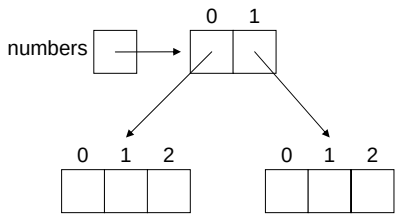


Variables hold the value `null` until we create the object

Creating the array

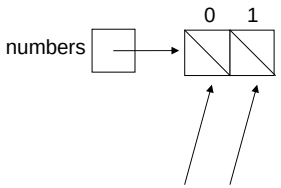
```
String[][] words;    // declare the array

words = new String[100][25]; //create the array
```

```
int[][] numbers;
numbers = new int[2][3];
```



Creating an array (alternate)

```
int[][] numbers;
numbers = new int[2][];
```



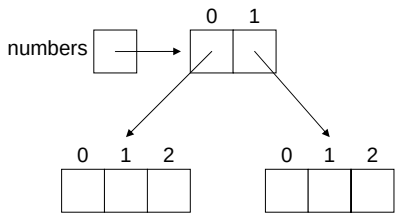
Question: What is the type of these elements?

```
numbers[0]
numbers[1]
```

Creating an array (alternate)

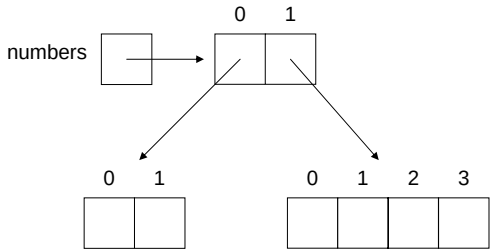
```
int[][] numbers;
numbers = new int[2][];
```

```
numbers[0] = new int[3];
numbers[1] = new int[3];
```



Creating ragged arrays

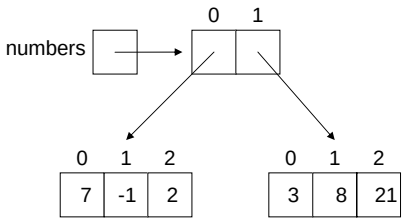
```
int[][] numbers;
numbers = new int[2][];
numbers[0] = new int[2];
numbers[1] = new int[4];
```



Initialising a 2d array

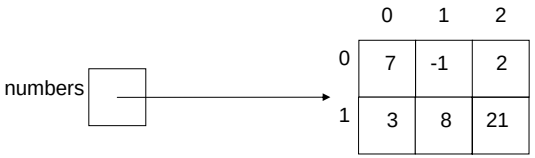
```
int[][] numbers;
numbers = new int[2][3];
numbers[0][0] = 7;
numbers[0][1] = -1;
numbers[0][2] = 2;
numbers[1][0] = 3;
numbers[1][1] = 8;
numbers[1][2] = 21;
```

```
int[][] numbers = { {7, -1, 2} , {3, 8, 21} };
```



Alternate visualisation

```
int[][] numbers = { {7, -1, 2} , {3, 8, 21} };
```



Nested Loops

One loop inside another

```
for (int i = 0; i < 3; i++) {
       for (int j = 0; j < 4; j++) {
              System.out.println(i + " " + j);
       }
}
```

```
for (int i = 0; i < 3; i++) {
       for (int j = 0; j < i; j++) {
              System.out.println(i + " " + j);
       }
}
```


Exercises

What is the output of the following code?

```
int SIZE = 6;
for (int i = 0; i < 6; i++) {
       for (int j = 0; j < i; j++) {
              if (j == 0 || j == i-1 || i == SIZE-1)
                     System.out.print("*");
              else
                     System.out.print("-");
       }
       System.out.println();
}
```

Given a one-dimensional array of integers, write code to test whether there exist any repeated numbers in that array.