ARRAY BASICS

zyBook 6.1, zyBook 6.2, zyBook 6.3, zyBook 6.4
Oracle - Java Tutorial: Arrays

@ Dr. Jessica Young Schmidt and NCSU Computer Science Faculty

HOW WOULD WE SOLVE THIS PROBLEM?

- Prompt user for number of students
- Prompt user for Project 1 grades for each student
- Print average grade for Project 1
- Print number of students with Project 1 grade higher than average

```
How many students? 5
Student 1's Project 1 Grade: 97
Student 2's Project 1 Grade: 92
Student 3's Project 1 Grade: 80
Student 4's Project 1 Grade: 99
Student 5's Project 1 Grade: 87
Average Project 1 Grade = 91.0
3 students were above average.
```

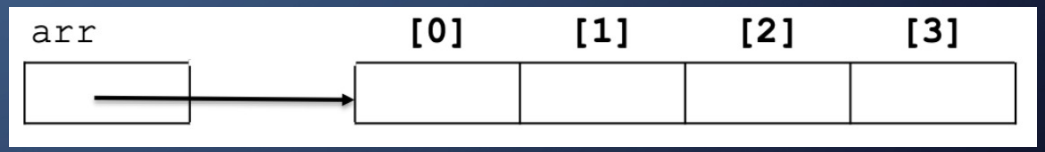
WHY IS THIS PROBLEM IS DIFFICULT?

```
How many students? 5
Student 1's Project 1 Grade: 97
Student 2's Project 1 Grade: 92
Student 3's Project 1 Grade: 80
Student 4's Project 1 Grade: 99
Student 5's Project 1 Grade: 87
Average Project 1 Grade = 91.0
3 students were above average.
```

- We need each input value twice:
  - to compute the average (a cumulative sum)
  - to count how many were above average
- We could read each value into a variable… but we:
  - do not know how many students are needed until the program runs
  - do not know how many variables to declare
- We need a way to declare many variables in one step.

WHAT ARE ARRAYS?

- Array → an indexed collection of variables of the same type
  - "An array is a simple but powerful programming language construct used to group and organize data."
- Elements → variables stored in an array
- Index → an integer indicating the position of a value in array
  - Arrays use zero-based indexing
  - What object have we already used that had zero-based indexing?
    - Strings



ARRAY USE EXAMPLE – STORING GRADES

- Option 1 – individual Project 1 variables

```
int project1student1;
int project1student2;
…
int project1student100;
```

- Option 2 – Project 1 array

```
int[] project1;
```

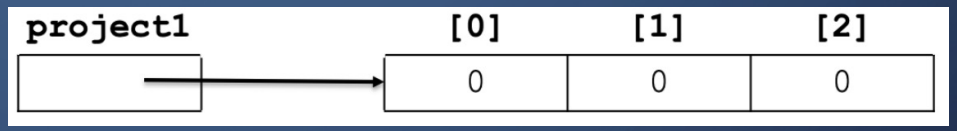
CONSTRUCTING ARRAYS

- Arrays are objects so must be constructed
- Syntax

  <type >[] <name > = new <type >[<size >];

- Project 1 Example

  int[] project1 = new int[3];



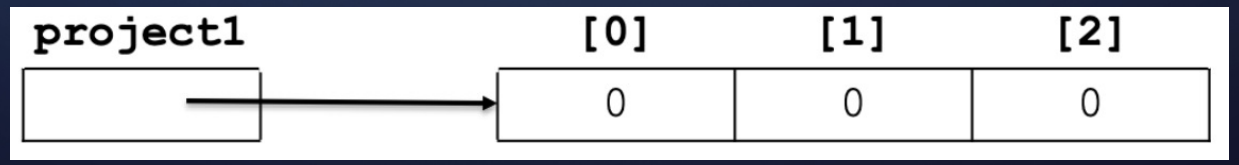
- Auto-initialization: The initialization of variables to a default value.

Type	Value
int	0
double	0.0
char	'\0'
boolean	false
Objects	null (Java keyword signifying no object)

GRADEBOOK

Create an int array to store the project one grades for 3 students (using class constant).

```
public class Gradebook {
    public static final int NUM_STUDENTS = 3;

    public static void main(String[] args) {
        int[] project1 = new int[NUM_STUDENTS];
    }
}
```



ARRAY CONVENTIONS

- Accessing an array element
  - Syntax

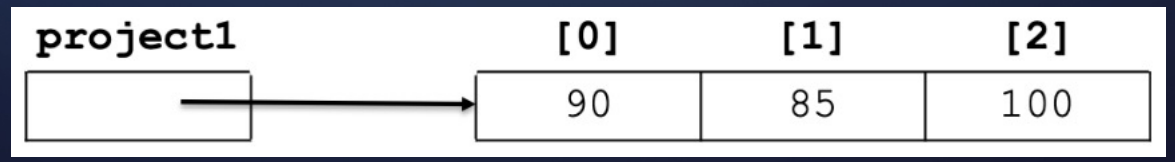
    <array-name> [<integer expression>]
  - Project 1 Example:

    project1 [2] // 100

- Getting the length of an array
  - Syntax

    <array-name>.length
  - Project 1 Example

    project1.length // 3



ARRAY CONVENTIONS

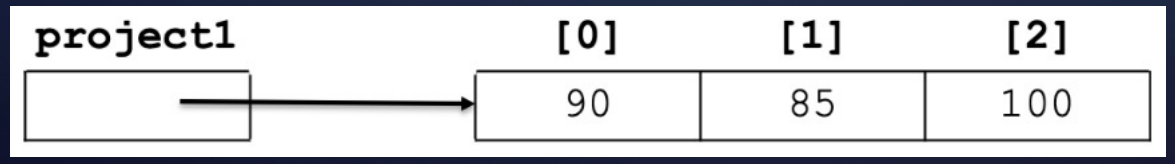
- Array traversal
  - Syntax

```
for (int i = 0; i < <array-name>.length; i++){
     <do something with <array-name>[i]>
}
```

  - Project 1 Example

```
for (int i = 0; i < project1.length; i++){
     System.out.println(i + ": " + project1[i]);
}
```

```
0: 90
1: 85
2: 100
```



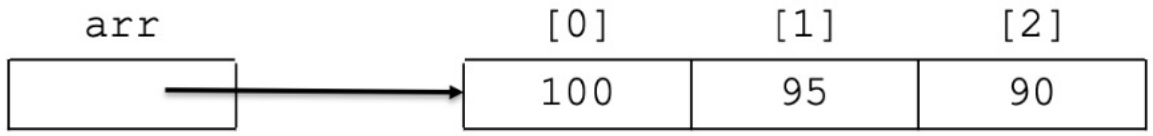
INITIALIZING ARRAYS: ARRAY TRAVERSAL

Syntax:

```
for (int i = 0; i < <name>.length; i++){
        <name>[i] = <value>;
}
```

Example:

```
int[] arr = new int[3];
for (int i = 0; i < arr.length; i++){
        arr[i] = 100 - 5 * i;
}
```



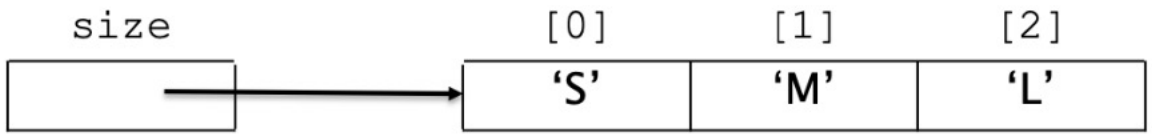
INITIALIZING ARRAYS: SHORTHAND

Syntax:

```
<type>[] <name> = {<value>, <value>,... <value>};
```

Example:

```
char[] size = {'S', 'M', 'L'};
```



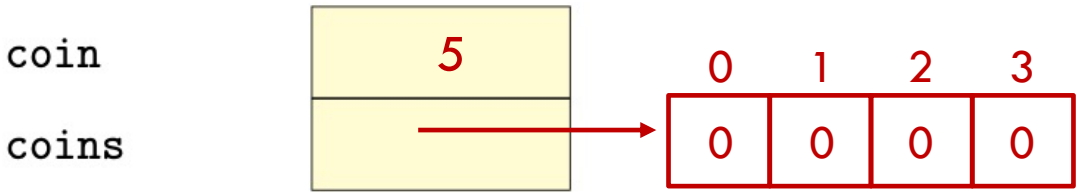
Using a loop to initialize arrays

```
import java.util.Scanner;

public class Gradebook {
    public static final int NUM_STUDENTS = 3;

    public static void main(String[] args) {
        int[] project1 = new int[NUM_STUDENTS];

        // Read in grades
        Scanner in = new Scanner(System.in);
        for (int i = 0; i < project1.length; i++) {
            System.out.print("Project 1 grade for Student " + i + ": ");
            while (!in.hasNextInt()) {
                in.next();
                System.out.print("Project 1 grade for Student " + i + " (as integer): ");
            }
            project1[i] = in.nextInt();
        }

        // Print grades
        System.out.println("\nProject 1 Grades:");
        for (int i = 0; i < project1.length; i++) {
            System.out.println("Student " + i + ": " + project1[i]);
        }
    }
}
```

```
$ java -cp bin Gradebook
Project 1 grade for Student 0: one
Project 1 grade for Student 0 (as integer): 90
Project 1 grade for Student 1: 85
Project 1 grade for Student 2: 100

Project 1 Grades:
Student 0: 90
Student 1: 85
Student 2: 100
```

EXAMPLE – COINS

```
int coin = 5;
int[] coins = new int[4];
for(int i = 0; i < coins.length; i++){
   coins[i] = 5 - i;
}
```



i = 0

0	1	2	3
5	0	0	0

i = 1

0	1	2	3
5	4	0	0

i = 2

0	1	2	3
5	4	3	0

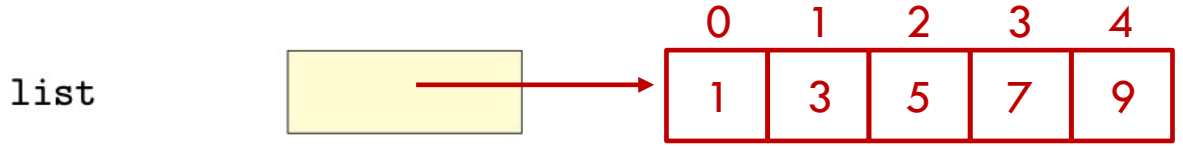
i = 3

0	1	2	3
5	4	3	2

i = 4

ARRAY CONTENTS

```
public class ArrayContent {

    public static void main(String[] args) {
        int[] list = new int[5];
        for (int i = 0; i < list.length; i++) {
            list[i] = 2 * i + 1;
        }

        System.out.println("First: " + list[0]);
        System.out.println("Middle: " + list[list.length / 2]);
        System.out.println("Last: " + list[list.length - 1]);
    }
}
```




```
$ java -cp bin ArrayContent
First: 1
Middle: 5
Last: 9
```

ARRAY ELEMENTS

Array elements can be treated as variables of a given type

```
int x = 0;
x = 3;
x++;
x--;
x *= 2;

int[] list = new int[3];
list[1] = 3;
list[2]++;
list[0]--;
list[0] *= 2;
```

ARRAY INDEX OUT OF BOUNDS

- Trying to access an element of an array at an index less than 0 or greater than length – 1 will result in an ArrayIndexOutOfBoundsException.
- Given an array of length 10 (int[] list = new int[10];), the following calls would result in ArrayIndexOutOfBoundsException being thrown:
  - list[-3] = 15;
  - int x = list[27];

EXAMPLE PROBLEM

- Prompt user for number of students
- Prompt user for Project 1 grades for each student
- Print average grade for Project 1
- Print number of students with Project 1 grade higher than average

```
How many students? 5
Student 1's Project 1 Grade: 97
Student 2's Project 1 Grade: 92
Student 3's Project 1 Grade: 80
Student 4's Project 1 Grade: 99
Student 5's Project 1 Grade: 87
Average Project 1 Grade = 91.0
3 students were above average.
```

```java
import java.util.Scanner;
public class Project1Gradebook {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("How many students?");
        while (!in.hasNextInt()) {
            in.next();
            System.out.print("How many students? (as int) ");
        }
        int numStudents = in.nextInt();

        int[] project1 = new int[numStudents];

        for (int i = 0; i < project1.length; i++) {
            System.out.print("Student " + (i + 1) + "'s Project 1 Grade: ");
            while (!in.hasNextInt()) {
                in.next();
                System.out.print("Student " + (i + 1) + "'s Project 1 Grade: ");
            }
            project1[i] = in.nextInt();
        }

        double average = 0;
        for (int i = 0; i < project1.length; i++) {
            average += project1[i];
        }
        average /= project1.length;
        System.out.println("Average Project 1 Grade = " + average);

        int count = 0;
        for (int i = 0; i < project1.length; i++) {
            if (project1[i] > average) {
                count++;
            }
        }
        System.out.println(count + " students were above average.");
    }
}
```