Arrays

• Objectives:
  – Discuss arrays
    • Syntax
    • Multi-dimensional arrays
Arrays

• Definition
  – Variables that store many objects of the same class.
  – access elements using brackets [ ]
  – Their indices start at 0
  – They are created using new
  – Their length must be specified.
Arrays

• Declaration for an array of integers:
  \[
  \text{int[]} \ a;
  \]
  – \text{int[]} : \text{indicates that} \ a \ \text{is an array of integers.}
  – \text{The declaration does not allocate any memory to contain the array elements.}

• Instantiate the array with \textit{new} operator.
  \[
  \text{public static void main(String[]} \ \text{args)}
  \{ \n    \text{int[]} \ a = \text{new int[10]}; \n    a[0] = 5; \ \text{a[1] = 10; \ a[2] = 15; \ a[3] = 20; \ a[4] = a[3]} \times 2; \ \ldots
  \}
  \]
Arrays

- **array length**
  - Arrays have a length = # of members
  - access using dot operator

```java
public static void main(String[] a)
{
  int[] squares = new int[5];
  for (int j = 0; j < squares.length; j++) {
    squares[j] = j * j;
    System.out.println("[j] = " + squares[j]);
  }
}
```


Arrays

- array element type
  - Arrays can be of any type or class
  - all elements in an array MUST have the same type or class

```java
float[] averages = new float[5];
int[] studentIds = new int[7];
byte[] byteArray = new byte[20];
char[] grades = new char[30];
```
Arrays

• subscript errors
  – Array subscripts start at 0 and end at length -1
  – Invalid index generates runtime exception
  – not detected by compiler during compiler time
  – Typical error:

```java
int[] a = new int[3];
a[1] = 15;
a[2] = 15;
a[3] = 15;  // error
```
Arrays

• **Initialization**
  
  – Array elements initialized to default values
    
    • zero for byte, short, int, long, float, double
    • null character for char
    • false for boolean
    • null for references

    int[] a = new int[5];    //set all elements to 0
Arrays

• **Initialization**
  – Arrays can be given initial values
  – array length defined by number of initializers
  – comma separated list enclosed in braces

```java
int[] a = new int[] { 7, -2, 9, 0, 19 };
```
Arrays

• Multidimensional
  – Two arrays
    
    int[] matrix0 = new int[4]  and
    int[] matrix1 = new int[4]
    
    Can be expressed as
    
    int[][] matrix = new int[2][4]
  
  – each dimension uses separate set of brackets
  – subscripts begin at 0
  
    for (int i = 0; i < matrix.length; i++) //length is 2
    for (int j = 0; j < matrix[i].length; j++) //length is 4
      matrix[i][j] = i + j;
Arrays

- initializing 2-dim arrays

```plaintext
int[][] matrix =
{
    
    {1, 2, 0, 6},
    {0, 7, 8, 5}

};
```
Arrays

**summary**

- store many objects of the same class
- are created by `new`
- have length as the number of objects
- start with index 0