

Arrays: 1D & 2D

CPSC 1181 – O.O.

Jeremy Hilliker

Summer 2017

Langara.

THE COLLEGE OF HIGHER LEARNING.

Outline

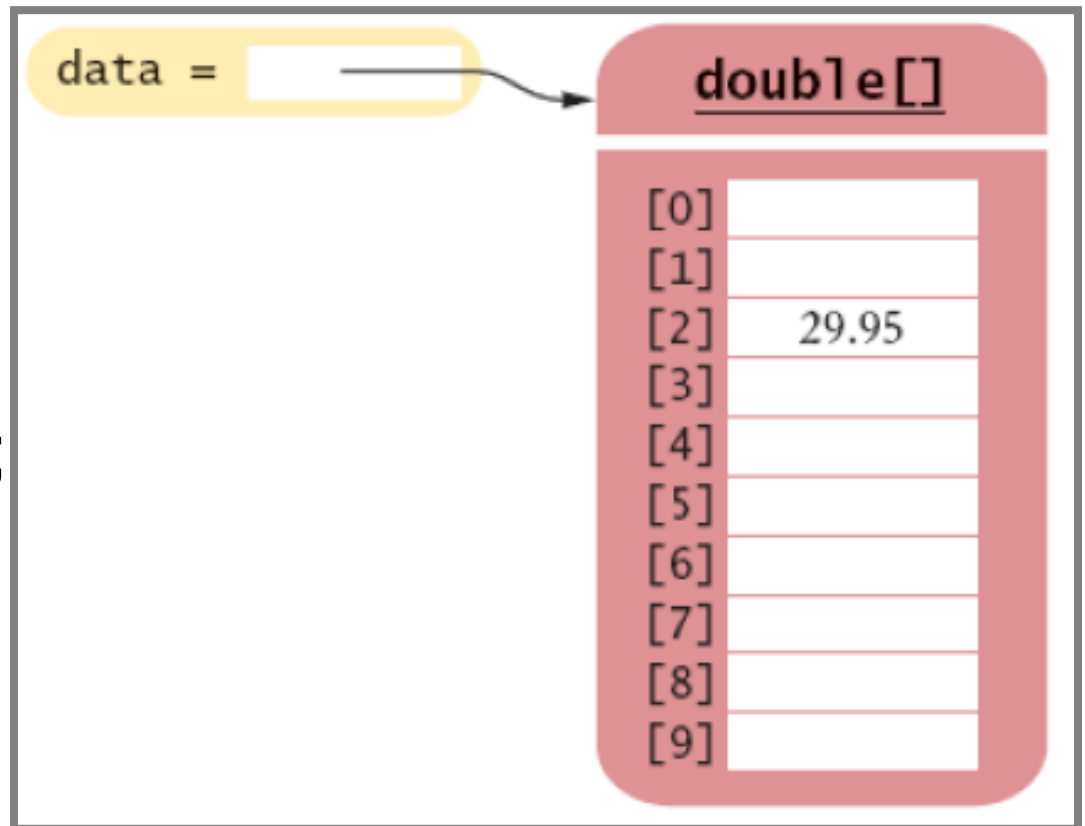
- 1D arrays
- Enhanced for loop
- 2D arrays

1D Array

- An ordered sequence of values of the same type
- **Note:** an array is also a type!
 - The set of ordered sequences of some type
- In java, arrays are a special type of object
 - Where the `[]` operator has special purpose
 - Get array length as `data.length` (not a method)
 - Index values range from `0` to `length - 1`
 - Access outside of this range results in an exception
 - `ArrayIndexOutOfBoundsException`
 - Arrays have a fixed length

1D Array: Access

- Use [] to access an element
- `data[2] = 29.95`
- `System.out.
println(data[2]);`



1D Array of Primitives

- An ordered sequence of values of the same type

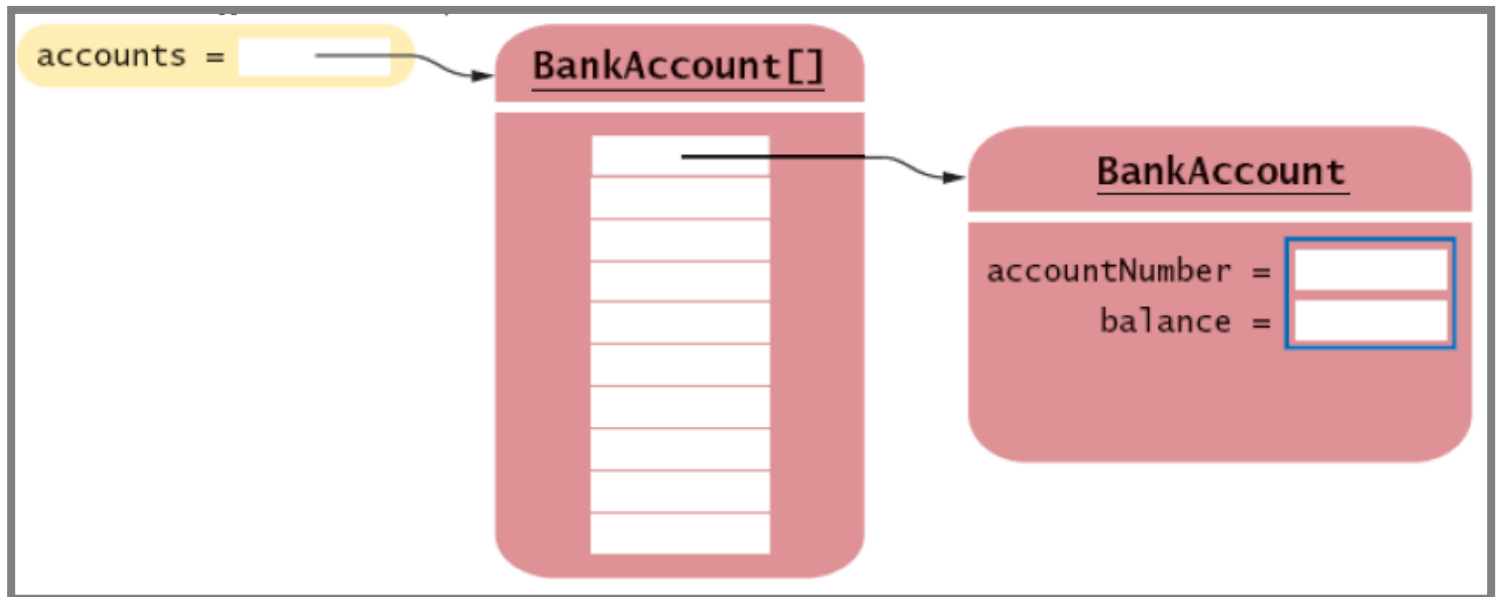
```
3      double[] d = new double[10]; // default 0.0
4      d[9] = 1/3.0
5
6      int[] i = new int[10];      // default 0
7      i[0] = Integer.MAX_VALUE;
8
9      boolean[] b = new boolean[2]; // default: false
10     b[0] = true;
11     b[1] = false;
```

1D Array of Objects

```
3      String[] names = new String[26]; // default: null
4      names[0] = "Alice";
5      names[1] = "Bob";
6      // ...
7
8      Integer[] is = new Integer[] { 0,1,2,3,4,5 }
```

1D Array of Objects

```
3 BankAccount[] accounts = new BankAccount[10];  
4 accounts[0] = new BankAccount();
```



Enhanced for Loop

```
2  ▼    public static void main(String[] args) {  
3  ▼        for(int i = 0; i < a.length; i++) {  
4          System.out.println(args[i]);  
5        }  
6        // "  
7  ▼    for(String s : args) {  
8        System.out.println(s);  
9    }  
10 }
```

Read as: “for each String s in args”

Q: can you use the enhanced for loop to make assignments in an array?

2D Arrays

- Think: rows & columns

```
4      char[][] board = new char[3][3];
5
6      board[0][0] = 'X';
7      board[1][2] = 'O';
8      board[i][j] = 'X';
```

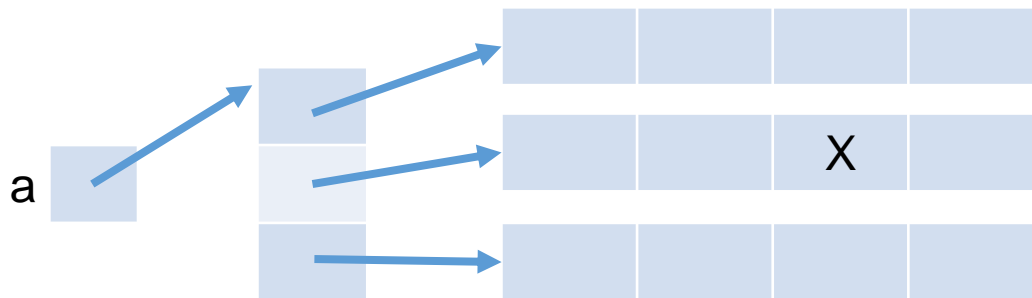
[i][j]	0	1	2
0	X		
1			O
2			

2D Arrays: Types

char[][]				
char[]	char	char	char	char
char[]	char	char	char	char
char[]	char	char	char	char
char[]	char	char	char	char

2D Arrays: Storage

```
char[][] a = new char[4];
```



What happens when we say:

```
a[1][2] = X;
```

?

Jagged Arrays

```
4      int[][] a = new int[5][];  // default: null
5      a[0] = new int[7];
6      a[1] = new int[2];
7      // a[2] = a[1];
8      a[3] = new int[5];
9      a[4] = new int[0];
```

int[][]							
int[]	int	int	int	int	int	int	int
int[]	int	int					
int[]	null						
int[]	int	int	int	int	int		
int[]	empty						

2D Array Traversal

```
4  ▾    for(int i = 0; i < ROWS; i++) {  
5  ▾        for(int j = 0; j < COLUMNS; j++) {  
6          System.out.print(board[i][j] + " ");  
7        }  
8        System.out.println();  
9    }
```

- Or use the enhanced for loop:

```
4  ▾    for(char[] row : board) {  
5  ▾        for(char c : row) {  
6          System.out.print(c + " ");  
7        }  
8        System.out.println();  
9    }
```

Example: Movie Ratings

Rows: Critics

Columns: Movies

Cells: Ratings

```
3  public class MovieRatings {
4
5      private final static int CRITICS = 3;
6  private final static String[] MOVIES =
7      {"Moon", "Jaws", "It", "They Live"};
8
9      private final float[][] ratings;
10
11  public MovieRatings() {
12      ratings = new float[CRITICS][MOVIES.length];
13      Random r = new Random(0);
14  for(int i = 0; i < ratings.length; i++) {
15  for(int j = 0; j < ratings[i].length; i++) {
16      ratings[i][j] = 1 + rand.nextInt(7) / 2f;
17  }
18  }
19  }
```

```

21 public int favMovie(int critic) {
22     int favMovie = 0;
23     final float[] row = ratings[critic];
24     for(int i = 0; i < row.length; i++) { // pattern: max
25         float rate = row[i]; // a rating
26         if(rate > row[favMovie]) { // if this movie is better
27             favMovie = i;
28         }
29     }
30 }

31
32 public int moviesBestCritic(String movieName) {
33     int m = getMovieIndex(movieName);
34
35     int bestCritic = 0;
36     for(int i = 0; i < ratings.length; i++) { // patter: max
37         // q: what's the rating if critic didnt review
38         if(ratings[i][m] > ratings[bestCritic][m]) {
39             bestCritic = i;
40         }
41     } // q: could we use the enhanced for loop?
42     return bestCritic;
43 }

```

```
45  ✓ public int getMovieIndex(String movie) {  
46  ✓     for(int i = 0; i < MOVIES.length; i++) { // pattern: finder  
47  ✓         if(MOVIES[i].equals(movie)) {  
48             return i;  
49         }  
50     } // q: could we use the enhanced for loop?  
51     return -1;  
52 }
```


Recap

- Arrays are objects (so they have a type)
- 1D Arrays
 - Primitives
 - Objects
- Enhanced for loop
- 2D arrays
 - Types
 - Storage
 - Jagged