



CS 149

Professor: Kevin Molloy

(adapted from slides originally developed by Alvin Chao)



Arrays

- Arrays are collections of the same type of element.
We can have arrays of type int, double, char, or String
- Declaration
 - Syntax: `BaseType[] ArrayName;`
 - Example: `int[] score;`
- Memory Allocation:
 - Syntax: `ArrayName = new BaseType[Length];`
 - Example: `score = new int[10];`
- Accessing Elements:
 - Syntax: `ArrayName[Index]`
 - Example: `score[5] = 8;`



Multiple uses of []

Uses of []:

- To declare that a variable is an array (of particular type)
- To indicate the amount of memory to allocate
- An operator to access an element of an array

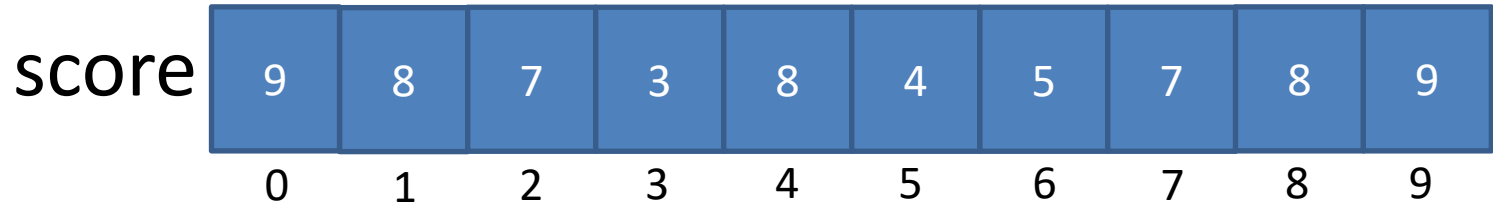
An Example:

```
int[] i;           // i is declared to be an array of int
i = new int[5];   // Memory is allocated for 5 int values
i[0] = 10;        // Assign 10 to the 0th element of i
```

```
System.out.printf("%d\n", i[0]); // Print the 0th element of i
```

Array Parameters

- Array state diagram



Formal declaration parameters examples:

- `public static void countPopular(int[] votes);`
 - `public static void main(String[] args);`
-
- Actual parameters passed
- Example:
- `countPopular(myVotes);`



Loops and Arrays

- Example: We declare an integer array score to hold bowling scores for each frame.

- What is stored in each element?

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



Array Length vs. String length

- `score.length` returns the length of the array.
- Notice this is different from a String length
- `str.length()` we have the `()` because string length is a method call to the String class versus `score.length` as an attribute.



- **Acknowledgements**

Parts of this activity are based on materials developed by David Bernstein.

</end>