



Arrays

CSCI 201

Principles of Software Development

Jeffrey Miller, Ph.D.

jeffrey.miller@usc.edu



Outline

- Arrays
- Program

Arrays



- An array is a group of memory locations related by the fact that they all have the same name and the same type
- Arrays can be addressed using brackets with the index (or position number) inside the brackets
 - › The first element of an array is in the 0th position
 - › So arrays are addressed from 0 to one less than the length of the array
- Each element in an array can be used just as the other scalar variables we have discussed
- To define an array, you have to use the “new” operator

```
int nums[] = new int[10];  
char names[] = new char[20];
```

Array Sample Code



```
1  public class ArrayClass {
2      public static void main(String [] args) {
3          double darr[] = new double[4];
4          darr[0] = 10.1;
5          darr[1] = 11.2;
6          darr[2] = 12.3;
7          darr[3] = 13.4;
8          System.out.print(darr[0]);
9          Scanner scan = new Scanner(System.in);
10         darr[2] = scan.nextDouble();
11         System.out.println(darr[2]);
12     }
13 }
```

Symbolic Constants



- Often array sizes are specified in constants that can be used throughout the program
- To define a constant, make the variable

```
public static final
```

```
1 public class Constants {
2     public static final int SIZE = 10;
3     public static void main(String [] args) {
4         int numArray[] = new int[SIZE];
5         for (int i=0; i < SIZE; i++) {
6             numArray[i] = 1;
7         }
8     }
9 }
```

Passing Arrays to Methods



- When defining the method, the size of the array is omitted because we can get the size from the array itself
- When calling the method, the brackets are left off completely
- Arrays are not primitive variables, so they are passed by reference

```
1 public class ArrayTest {
2     public static void myMethod(int numArr[]) {
3         int sizeOfArray = numArr.length;
4     }
5     public static void main(String [] args) {
6         int numArray[] = new int[10];
7         myMethod(numArray);
8     }
9 }
```

Array Example



```
1 public class ArrayTest {
2     public void addToIndex(float myArray[], int index, float amount) {
3         myArray[index] += amount;
4     }
5
6     public void printArray(float myArr[]) {
7         for (int count=0; count < myArr.length; count++) {
8             System.out.println("arr[" + count + "] = " + myArr[count]);
9         }
10    }
11
12    public static void main(String [] args)
13        float arr[] = new float[10];
14        for (int i=0; i < arr.length; i++) {
15            arr[i] = Math.random() * 20.0f;
16        }
17        ArrayTest at = new ArrayTest();
18        at.printArray(arr);
19        at.addToIndex(arr, 3, 20.4f);
20        at.printArray(arr);
21    }
22 }
```

Multi-Dimensional Arrays



- You are able to have multiple dimensions on an array instead of just one index

```
1     int my2DArray[][] = new int[10][20];
2     for (int i=0; i < my2DArray.length; i++) {
3         for (int j=0; j < my2DArray[i].length; j++) {
4             my2DArray[i][j] = 10;
5         }
6     }
```




Outline

- Arrays
- Program

Program



- Write a program that randomly generates dice rolls. The number of rolls will be provided by the user. Output the number of times each number occurred followed by the percentage. Here is a sample execution with user input bolded.

```
c:\>java csci201.Dice
How many rolls? 5000
The number 1 occurred 800 times (16%).
The number 2 occurred 750 times (15%).
The number 3 occurred 850 times (17%).
The number 4 occurred 825 times (16.5%).
The number 5 occurred 775 times (15.5%).
The number 6 occurred 800 times (16%).
c:\>
```