

Array Processing and Examples – Part I

Hui Chen

Department of Computer & Information Science

Brooklyn College

Objectives

- To program common array operations (§7.2.6)
 - displaying arrays,
 - summing all elements,
 - finding the minimum and maximum elements, and
 - random shuffling, and shifting elements.
- To simplify programming using the foreach loops (§7.2.7)

Processing Arrays

- Initializing arrays with input values
- Initializing arrays with random values
- Printing arrays
- Summing all elements
- Finding the largest element
- Finding the smallest index of the largest element
- Random shuffling
- Shifting elements

Initializing arrays with input values

```
java.util.Scanner input = new java.util.Scanner(System.in);  
System.out.print("Enter " + myList.length + " values: ");  
for (int i = 0; i < myList.length; i++) {  
    myList[i] = input.nextDouble();  
}
```

Initializing arrays with random values

```
for (int i = 0; i < myList.length; i++) {  
    myList[i] = Math.random() * 100;  
}
```

Printing arrays

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

Summing all elements

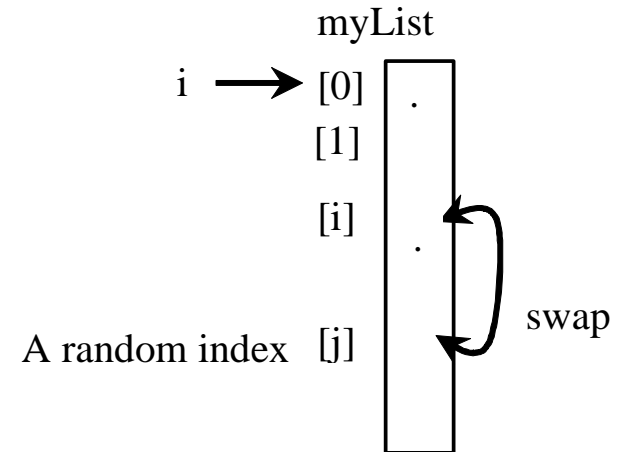
```
double total = 0;  
for (int i = 0; i < myList.length; i++) {  
    total += myList[i];  
}
```

Finding the largest element

```
double max = myList[0];  
for (int i = 1; i < myList.length; i++) {  
    if (myList[i] > max) max = myList[i];  
}
```


Random shuffling

```
for (int i = 0; i < myList.length - 1; i++) {  
    // Generate an index j randomly  
    int j = (int)(Math.random()  
        * myList.length);  
  
    // Swap myList[i] with myList[j]  
    double temp = myList[i];  
    myList[i] = myList[j];  
    myList[j] = temp;  
}
```

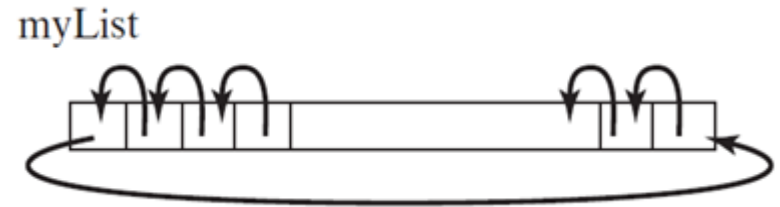


Shifting Elements

```
// retain the first element  
double temp = myList[0];
```

```
// shift elements left  
for (int=1; i<myList.length; i++) {  
    myList[i-1] = myList[i];  
}
```

```
// move the first element to fill in the last position  
myList[myList.length - 1] = temp;
```



Questions?

Enhanced For Loop

- JDK 1.5 introduced a new for loop that enables you to traverse the complete array sequentially without using an index variable. For example, the following code displays all elements in the array myList:

```
for (double value: myList) {  
    System.out.println(value);  
}
```

- In general, the syntax is

```
for (elementType value: arrayRefVar) {  
    // Process the value  
}
```

- You still have to use an index variable if you wish to traverse the array in a different order or change the elements in the array.

Questions?

- How many examples here can you rewrite using the enhanced for loop?