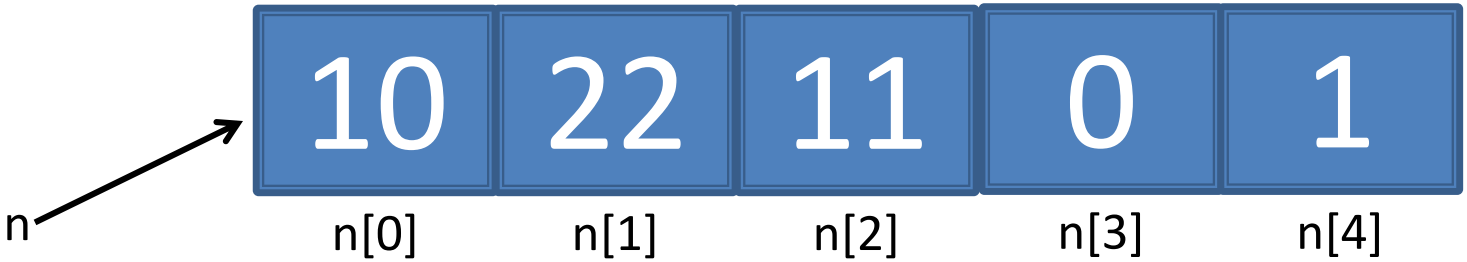
CSE 20
Intro to Computing I

Lecture 9 – Arrays (2)

Announcements

- Lab #11 this week
  - Due before your next lab
  - Make sure to show your work to **YOUR TA** (or me) before submission
- Project #2 out this Friday
  - Due 12/1 (Friday)
- Reading assignment
  - Chapter 5.1-5.5 of textbook (Due date extended)
- Want Extra Credit???

Array




```
int[] n = {10, 22, 11, 0, 1};
for (int i = 0; i < n.length; i++)
    System.out.println("Index " + i + " has value " + n[i]);
```

Index 0 has value 10
Index 1 has value 22
Index 2 has value 11
Index 3 has value 0
Index 4 has value 1

n.Length tells you have many elements are in n

Array : Pit Falls

```
int[] arr = new int[5];

for (int i = 0; i <= arr.length; i++)
 System.out.println("Index " + i + " is " + arr[i]);
}
```

Logical Error

i = 5

0	0	0	0	0
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

```
Exception in thread "main"
java.lang.ArrayIndexOutOfBoundsException: 5
at ErrorsTest.main(ErrorsTest.java:7)
```

Runtime Error (i = 5 but arr[5] does not exist)

SumAll Sequence

SumAll sums all the numbers from 1 to max:

0	1	2	3	4	5	6	7	8	9
0	**1**	**3**	**6**	**10**	**15**	**21**	**28**	**36**	**45**

```
n[0] = 0;
for (int i = 1; i <= max ; i++) {
    n[i] = n[i-1] + i;
}
```

What happens if we start at i = 0 instead of i = 1?

n[i] = n[i-1] + i // sub i = 0;

n[0] = n[0-1] + 0;
n[0] = n[-1] + 0;

SumAll - Alternatives

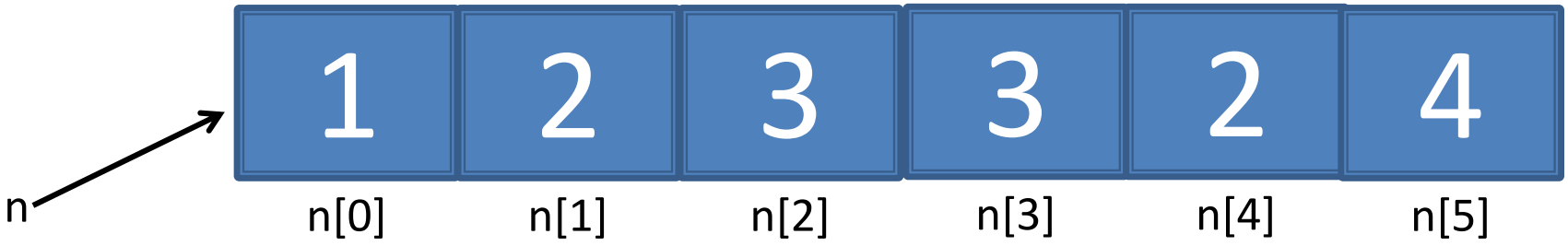
```
int i = 1;
n[0] = 0;

while (i <= max) {
 n[i] = n[i-1] + i;
 i++;
}

do {
 n[i] = n[i-1] + i;
 i++;
} while (i <= max);

do
 n[i] = n[i-1] + i;
while (++i <= max);
```

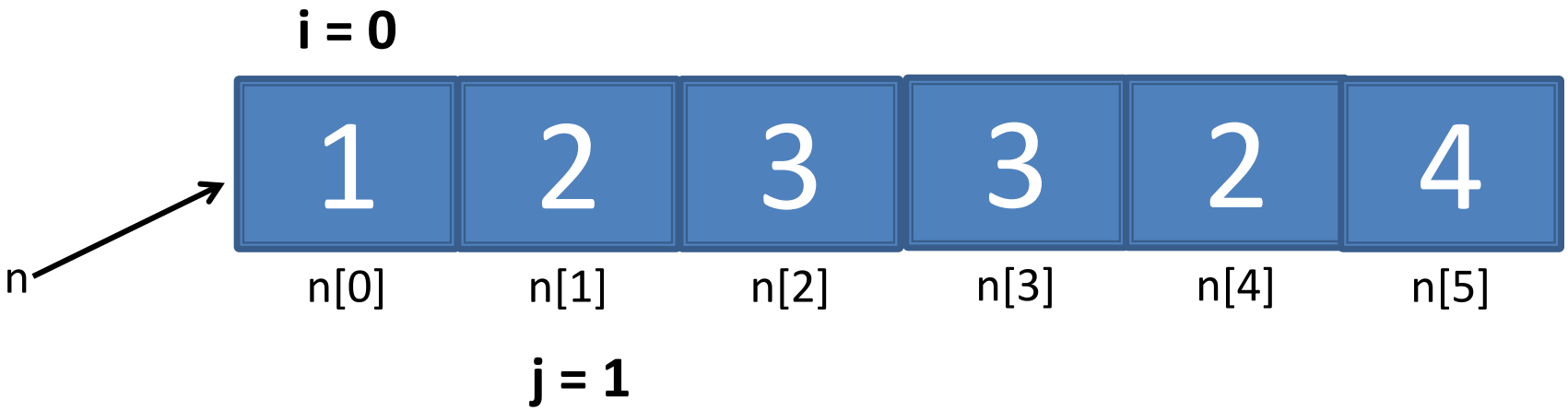
Array – compare elements



Nested Loop

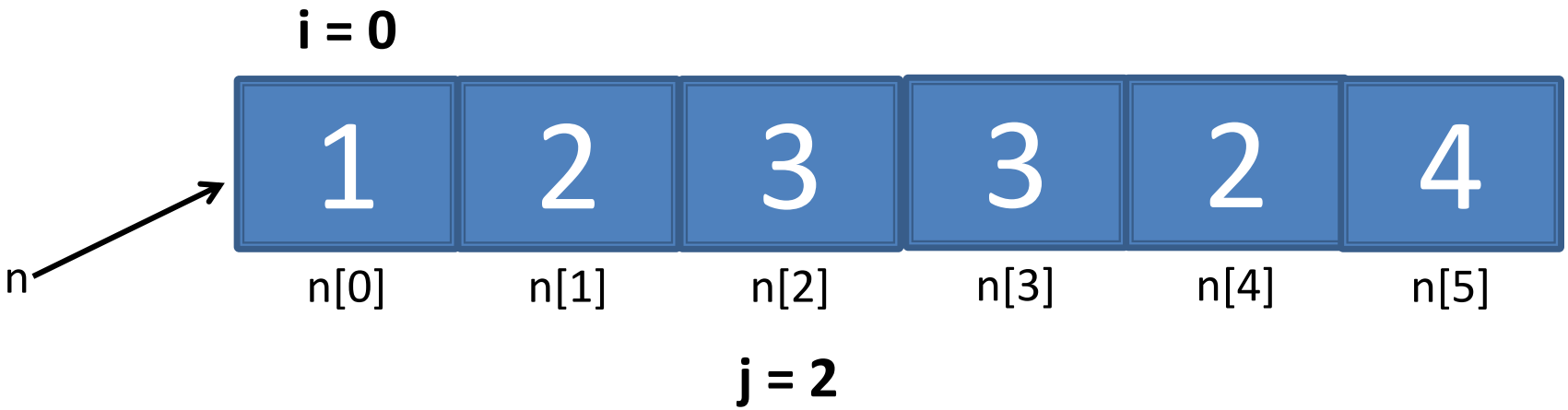
```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

Array – compare elements



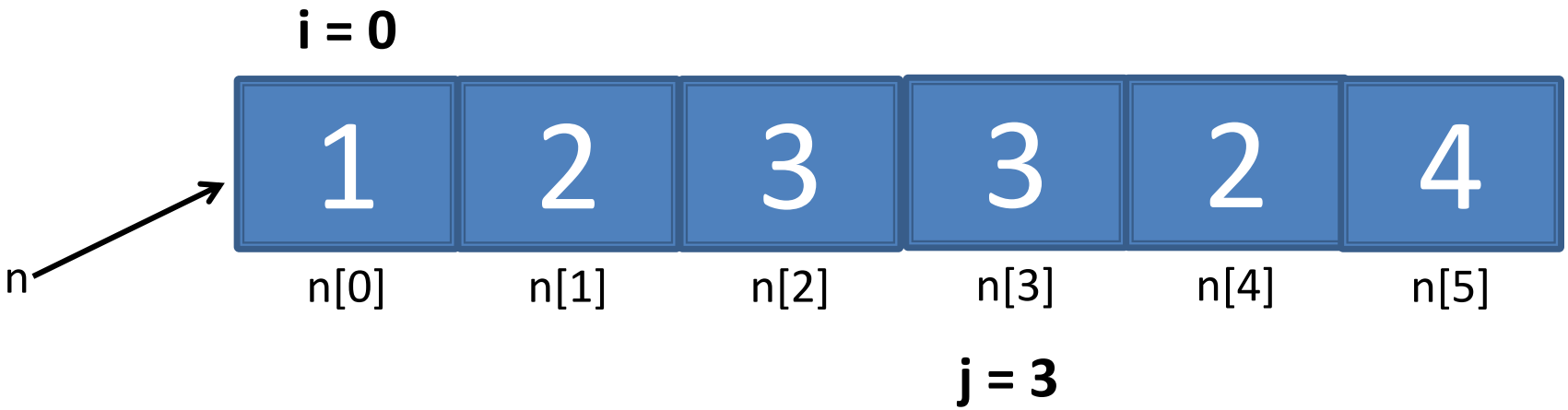

```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

Array – compare elements



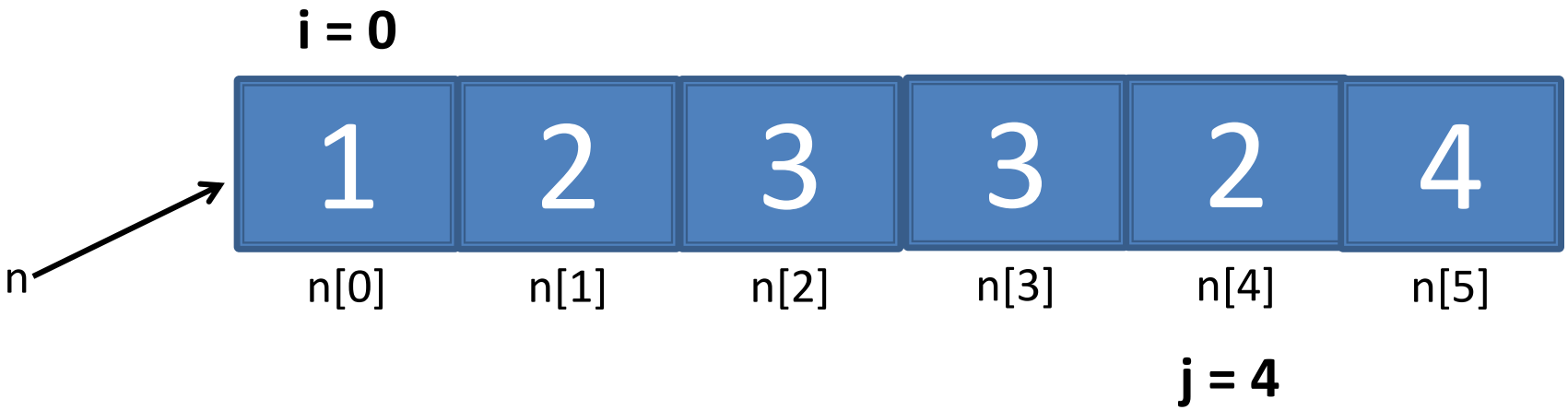

```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

Array – compare elements



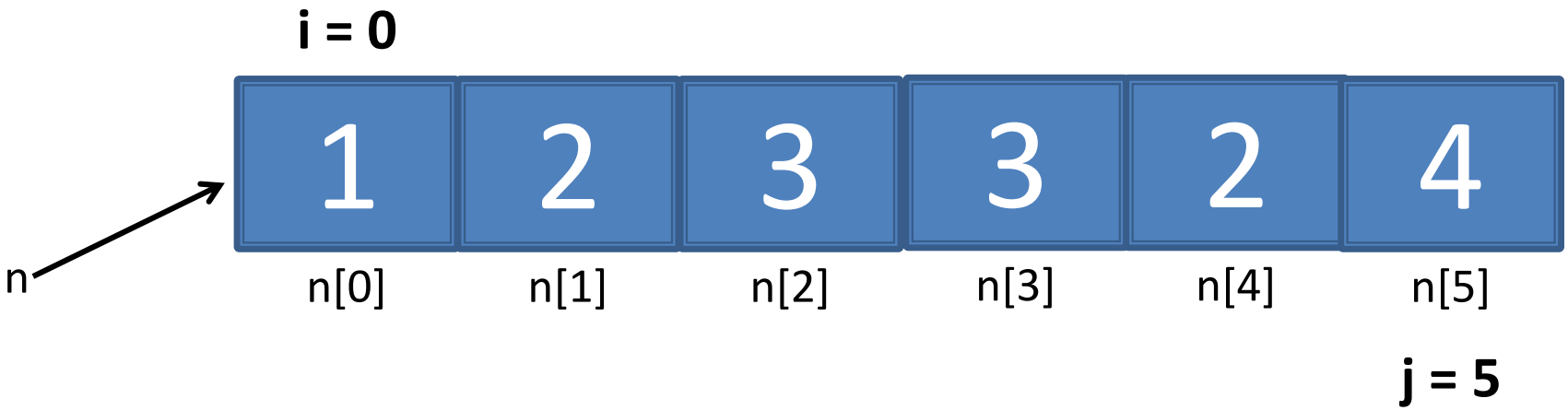

```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

Array – compare elements



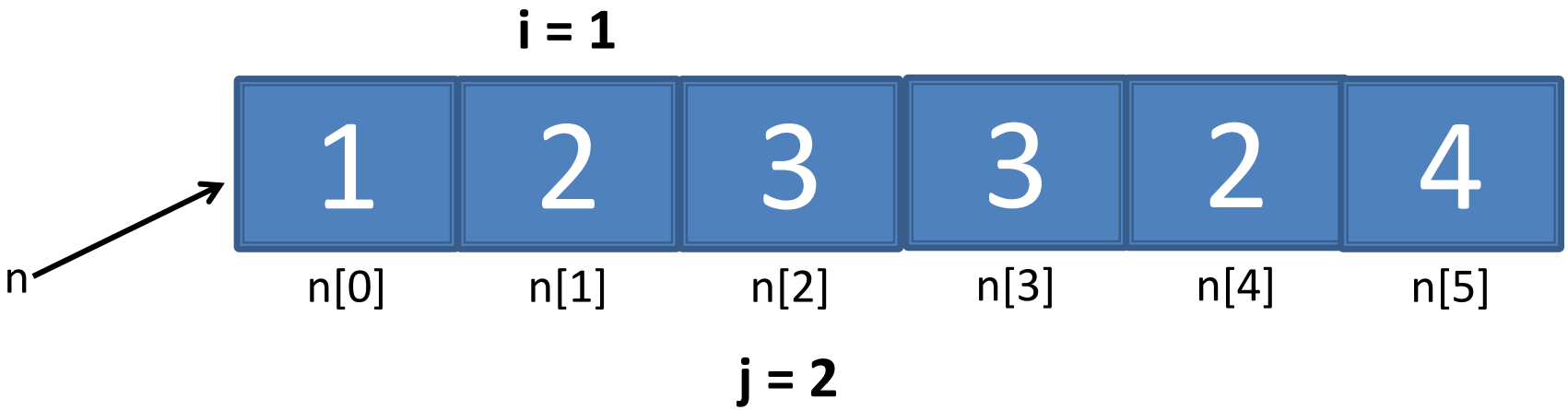

```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

Array – compare elements



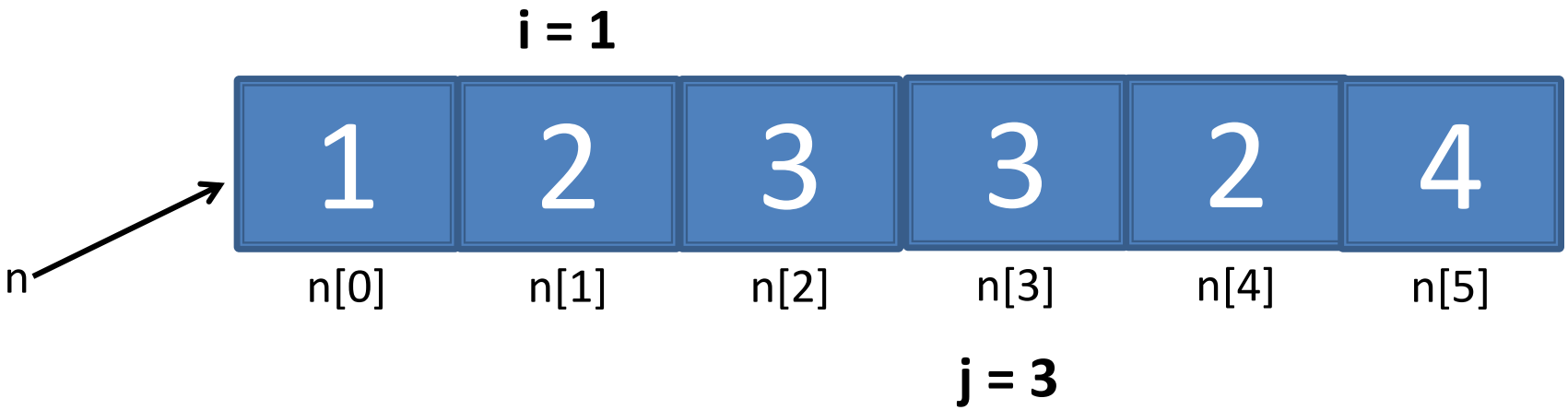

```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

Array – compare elements



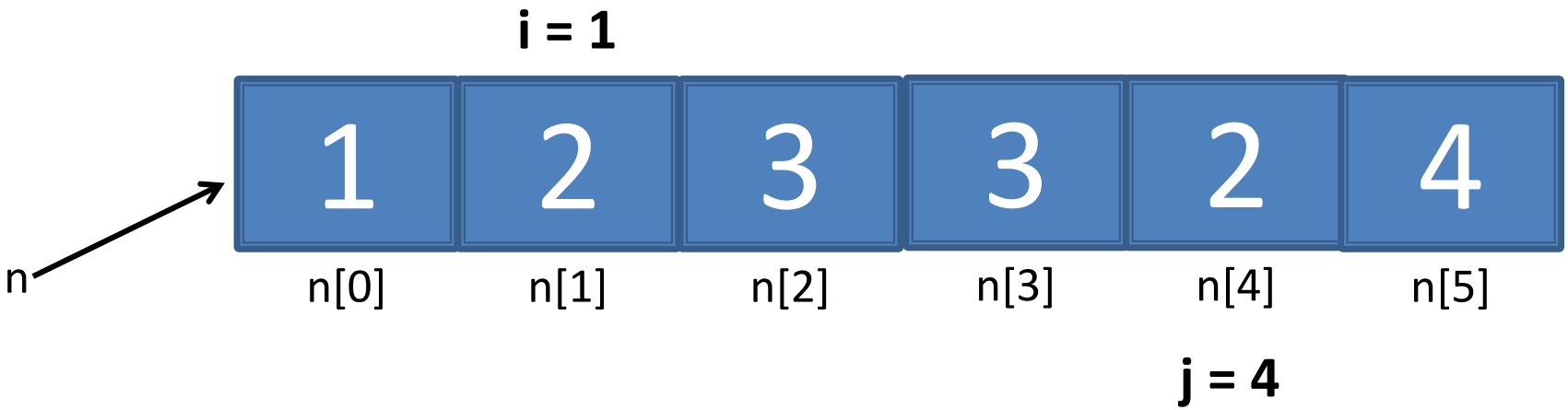

```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

Array – compare elements




```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

Array – compare elements

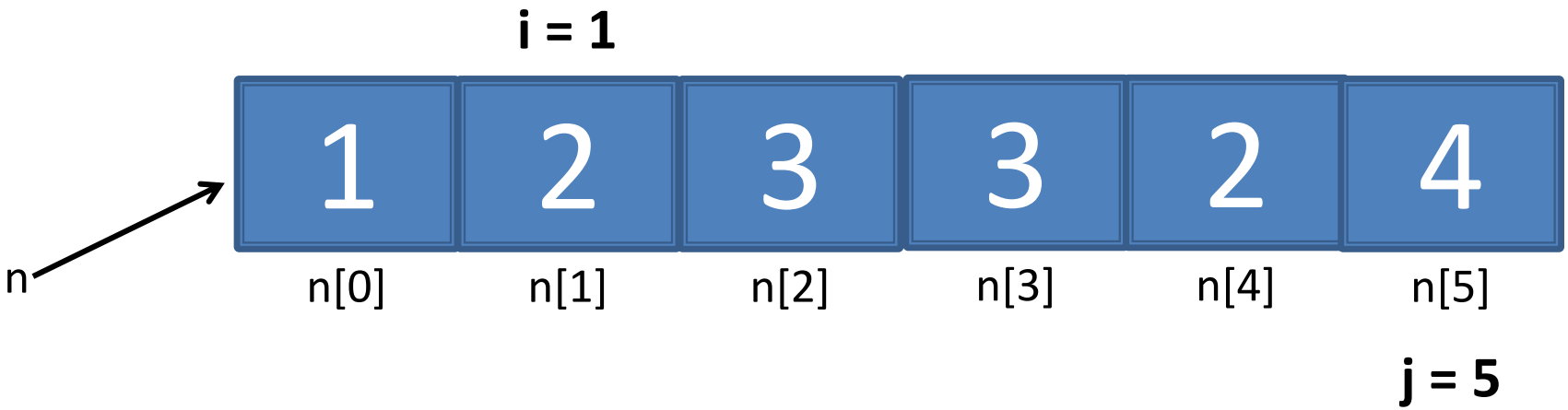



```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

OUTPUT:

1 is same as 4

Array – compare elements

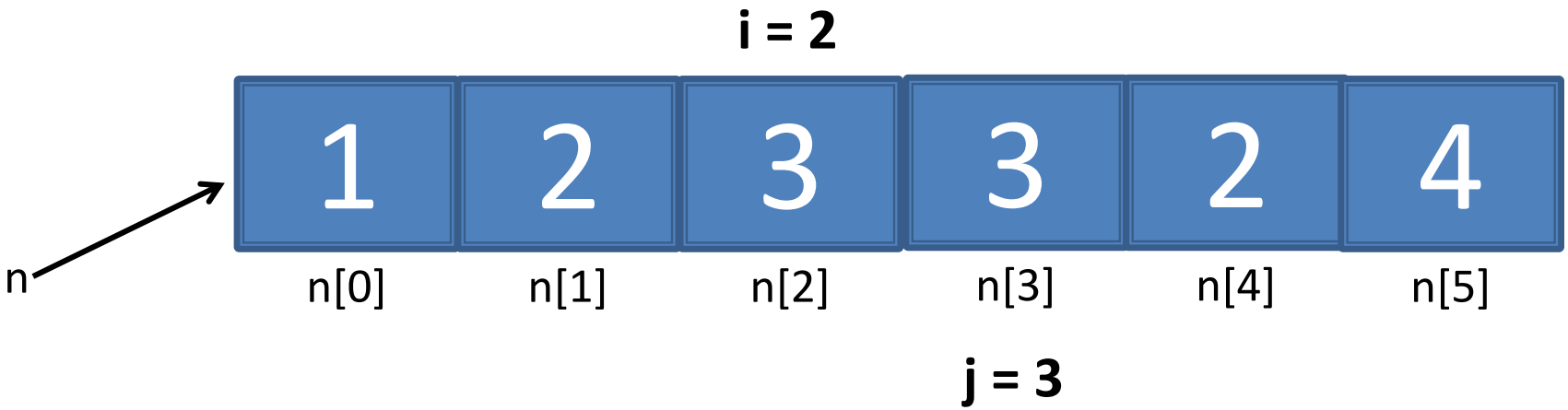



```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

OUTPUT:

1 is same as 4

Array – compare elements



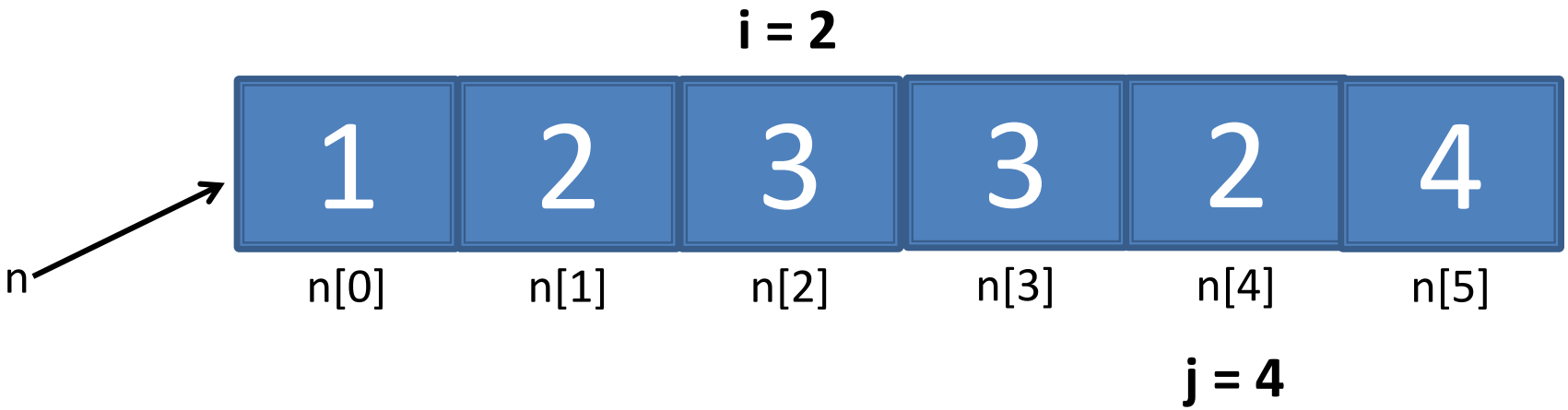

```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

OUTPUT:

1 is same as 4

2 is same as 3

Array – compare elements



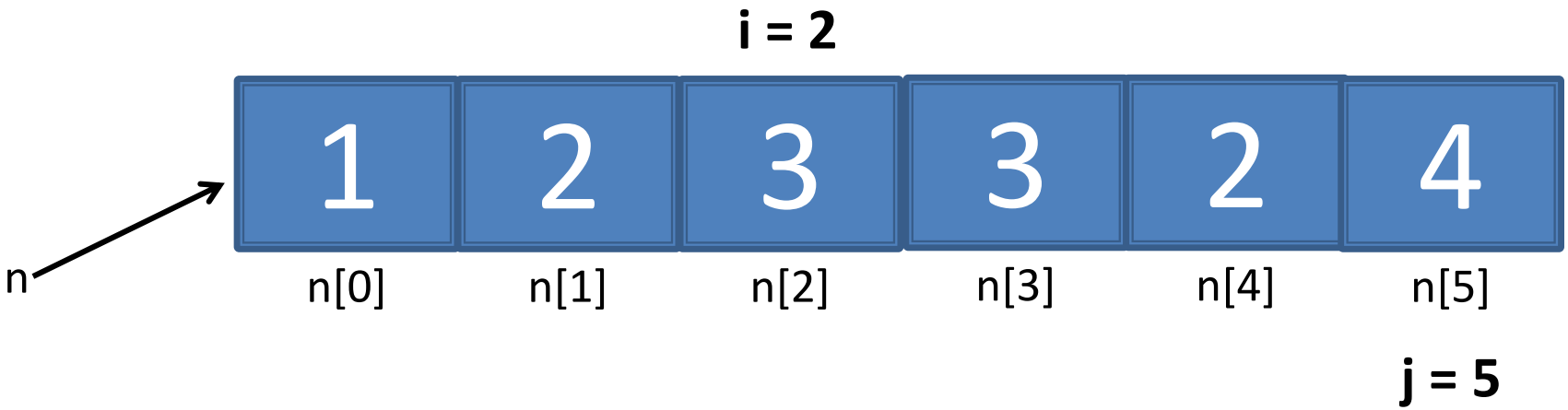

```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

OUTPUT:

1 is same as 4

2 is same as 3

Array – compare elements



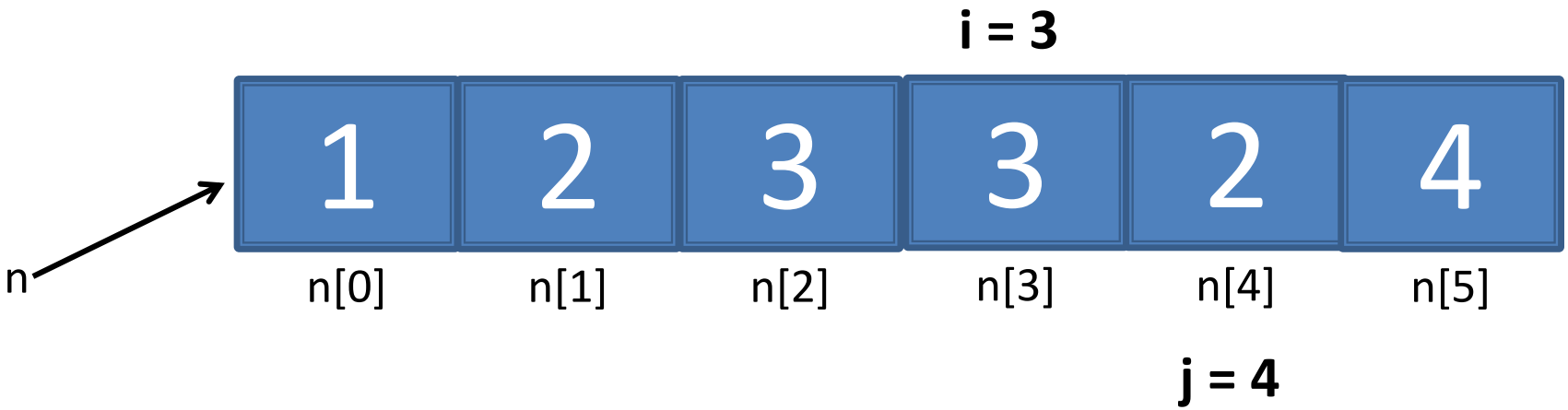

```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

OUTPUT:

1 is same as 4

2 is same as 3

Array – compare elements

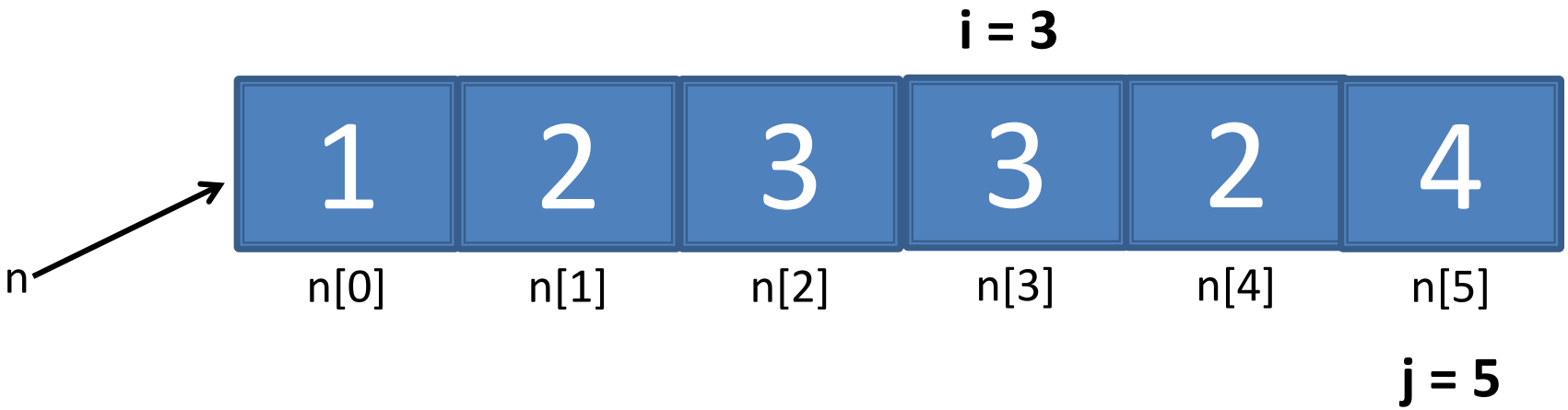



```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

OUTPUT:

1 is same as 4
2 is same as 3

Array – compare elements



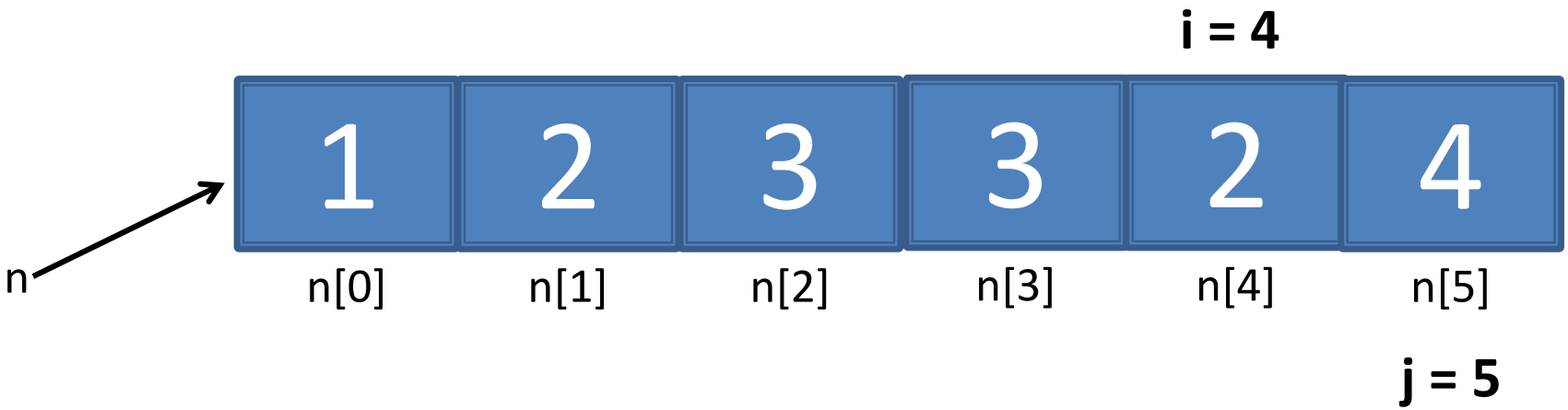

```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

OUTPUT:

1 is same as 4

2 is same as 3

Array – compare elements

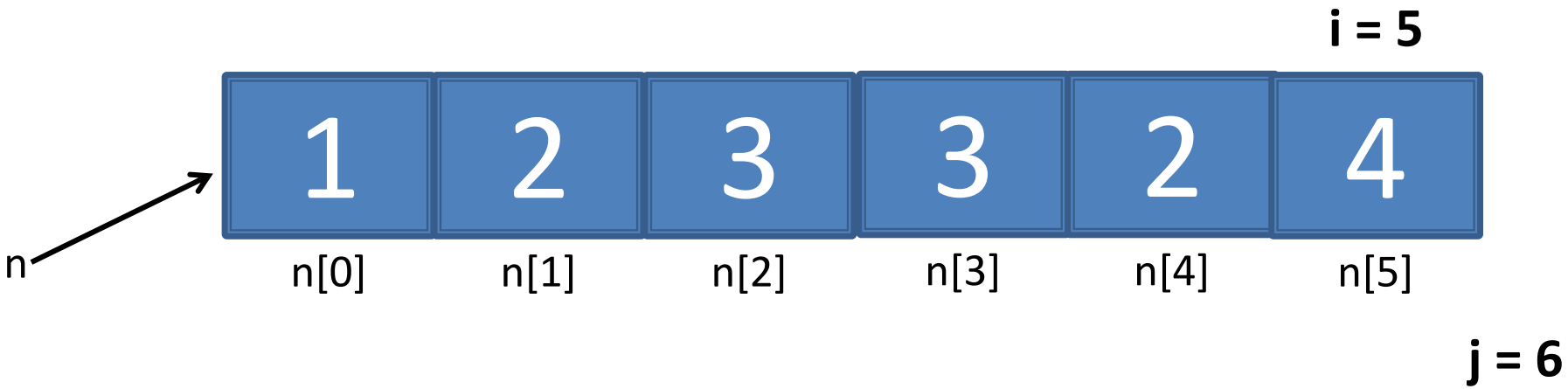



```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

OUTPUT:

1 is same as 4
2 is same as 3

Array – compare elements




```
for (i = 0; i < n.length; i++)
    for (j = i + 1; j < n.length; j++)
        if (n[i] == n[j])
            System.out.println(i + “same as ” + j);
```

OUTPUT:

1 is same as 4
2 is same as 3

Code : Scope

```
int sum = 0, i = 100;
int max = 10;

for(i = 0;i < max;i++) {
  System.out.print("Please enter " + i + "num:");
  int num = input.nextInt();
  sum += num;

  if (i % 3 == 0)
    System.out.println(num + " " + sum);
}

int num = 100;
System.out.println(num + " " + i);
System.out.println(sum);
```

2 num declared??

Sample Output

```
int sum = 0, i = 100;
int max = 10;

for(i = 0;i < max;i++) {
  System.out.print("Please enter " + i +
"num:");
  int num = input.nextInt();
  sum += num;

  if (i % 3 == 0)
    System.out.println(num + " " + sum);

}

int num = 100;
System.out.println(num + " " + i);
System.out.println(sum);
```

```
Please enter 0 number: 10
10 10
Please enter 1 number: 9
Please enter 2 number: 3
Please enter 3 number: 4
4 26
Please enter 4 number: 2
Please enter 5 number: 5
Please enter 6 number: 6
6 39
Please enter 7 number: 7
Please enter 8 number: 4
Please enter 9 number: 2
2 52
100 10
52
```

Array and Scope

```
int[] n = {1, 2, 3, 3, 2, 4};

for (int i = 0; i < n.length; i++) {

    for (int j = i + 1; j < n.length; j++)

        if (n[i] == n[j])

        System.out.println(i + " same as " + j);

    System.out.println(i + " and " + j);

}
```

j is undefined in this scope

DataAnalyze.java

- We are asked analyze data collected from 4 trials of experiments.
- Tasks:
  - Get sample size (number of samples in each trial) from user.

```
Please enter the sample size: 3
```

  - Get samples from user for each trial.

```
Enter numbers for Trial 0
Enter sample #0:50
Enter sample #1:49
Enter sample #2:51
…
```

  - Store sample of each trial in an array.

DataAnalyze.java

- Tasks:
  - Calculate the average of each trial.
  - Display the samples and average for each trial in a correct format.

Sample #	Trial 1	Trial 2	Trial 3	Trial 4
0	50	30	25	15
1	49	31	26	16
2	51	32	27	17
Average:	50	31	26	16

  - Find the minimum and maximum averages from the trials.
  - Display the minimum and maximum averages.

Min Average: 16
Max Average: 50

DataAnalyze.java

- Tasks:
  - Compare the minimum and maximum averages to see how closely the trials match.
    - Exactly - minimum and maximum averages are the same
    - Concur – max less than 2 x min
    - No concur – none of above
  - Display the result of the comparison

The trials do NOT concur!