

CSE 20

Intro to Computing I

Lecture 11 – Arrays (4)

Announcements

- ▶ Lab #12 this week
 - Due before your next lab
 - Make sure to show your work to **YOUR TA** (or me) before submission
- ▶ Project #2
 - Due 12/1 (Friday)
- ▶ Final Exam (12/11, Monday, 3pm)
 - Cover everything
 - Open notes
 - Review next lecture
- ▶ Extra Credit from zyBooks
 - Course evaluation online by 12/8

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);
```

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

Factorial: definition

$$n! = \begin{cases} 1, & n = 0 \\ n \times (n - 1) \times (n - 2) \dots \times 2 \times 1, & n > 0 \end{cases}$$

Handwritten diagram illustrating the factorial calculation for $n=5$:

$1 \times 2 \times 3 \times 4 \times 5 \dots \times \text{max}$

The diagram shows a sequence of numbers 1, 2, 3, 4, 5, followed by an ellipsis and the word "max", all enclosed in red boxes and connected by red multiplication signs. This represents the product of all integers from 1 to n .

Factorial

Diagram illustrating a memory array a for calculating factorials. The array is indexed from 0 to max . The values stored in the array are:

| Index | 0 | 1 | 2 | 3 | 4 | ... | ... | ... | max |
|-------|---|---|---|---|----|-----|-----|-----|-----|
| Value | 1 | 1 | 2 | 6 | 24 | | | | |

The array a is pointed to by the variable a .

$$a[p] = 1$$

1 ,

Fibonacci numbers

$$F_0 = 0$$

$$F_1 = 1$$

$$F_k = F_{k-1} + F_{k-2}$$

0, 1, 1, 2, 3, 5, 8, 13,

Fibonacci numbers

| 0 | 1 | 2 | 3 | 4 | 5 | | | |
|---|---|---|---|---|---|--|--|--|
| 0 | 1 | 1 | 2 | | | | | |

↑ $f[0] = 0, f[1] = 1$

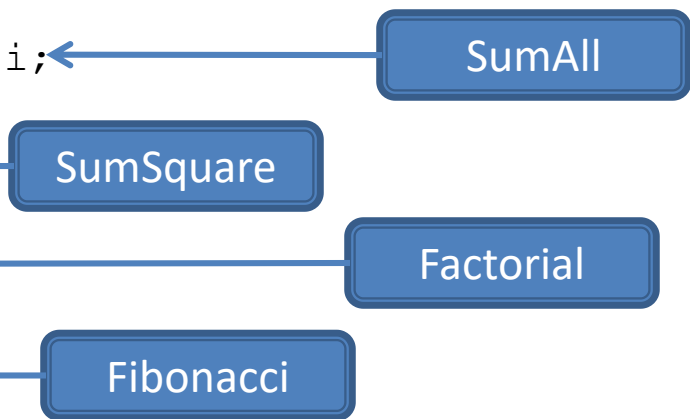
$i=2$ $f[2] = f[1] + f[0] = 1 + 0 = 1$

$i=3$ $f[3] = f[2] + f[1] = 1 + 1 = 2$

$i=n$ $f[n] = f[n-1] + f[n-2]$

Nested Structure: ChooseFunc.java

```
for (int i = 1; i <= max; i++) {  
    if (func == 1) {  
        resArr[i] = resArr[i-1] + i; ← SumAll  
    } else if (func == 2) {  
        // Fill-in (Sum Square) ← SumSquare  
    } else if (func == 3) {  
        // Fill-in (Factorial) ← Factorial  
    } else if (func == 4) {  
        // Fill-in (Fibonacci) ← Fibonacci  
    } else {  
        System.out.println("Error: Do not know " + func);  
        // Fill-in (Stop the for-loop from iterating)  
    }  
}
```



ChooseFunc.java: Scope

```
for (int i = 1; i <= max; i++) {  
    if (func == 1) {  
        resArr[i] = resArr[i-1] + i;  
    } else if (func == 2) {  
        // Fill-in (Sum Square)  
    } else if (func == 3) {  
        if (i == 1)  
            // Fill-in (Factorial)  
        else  
            // Fill-in (Factorial)  
    } else if (func == 4) {  
        if (i == 1)  
            // Fill-in (Fibonacci)  
        else  
            // Fill-in (Fibonacci)  
    } else {  
        System.out.println("Error: Do not know " + func);  
        // Fill-in (Stop the for-loop from iterating)  
    }  
}
```