

# CSCI 132:

# Basic Data Structures and Algorithms

2D Arrays, Program 1

Reese Pearsall  
Spring 2025

# Announcements

Program 1 posted  
→ Due Wednesday Feb 19

Next lab will be posted  
sometime soon



**java** when I have  
an atom of  
difference between  
my type and the  
expected type



Python when I  
cast a float  
into an  
unsigned Toyota  
Yaris 2023

## 2D Arrays

```
array = [ 0, 1, 2, 3, 4 ]
```

Here is a one-dimensional array that holds ints (`int[]`)

## 2D Arrays

```
array = [ 0, 1, 2, 3, 4 ]
```

Here is a one-dimensional array that holds ints (`int[]`)

```
array2 = [ [0, 1, 2, 3, 4],  
            [5, 6, 7, 8, 9],  
            [10, 11, 12, 13, 14],  
            [15, 16, 17, 18, 19],  
            [20, 21, 22, 23, 24] ]
```

Here is a two-dimensional array that holds ints (`int[][]`)

## 2D Arrays

```
array = [ 0, 1, 2, 3, 4 ]
```

Here is a one-dimensional array that holds ints (`int[]`)

```
array2 = [ [0, 1, 2, 3, 4],  
            [5, 6, 7, 8, 9],  
            [10, 11, 12, 13, 14],  
            [15, 16, 17, 18, 19],  
            [20, 21, 22, 23, 24] ]
```

```
array2[3]
```

Will return a 1D-array

Here is a two-dimensional array that holds ints (`int[][]`)

## 2D Arrays

```
array = [ 0, 1, 2, 3, 4 ]
```

Here is a one-dimensional array that holds ints (`int[]`)

```
array2 = [ [0, 1, 2, 3, 4],  
           [5, 6, 7, 8, 9],  
           [10, 11, 12, 13, 14],  
           [15, 16, 17, 18, 19],  
           [20, 21, 22, 23, 24] ]
```

```
array2[3][1]
```

Will return a singular  
element in that array

Here is a two-dimensional array that holds ints (`int[][]`)

## 2D Arrays

We can use a single **for** loop to print out each element of this array

```
array = [ 0, 1, 2, 3, 4 ]
```

```
for(int i = 0; i < array.length; i++) {  
    System.out.println(array[i]);  
}
```

Here is a one-dimensional array that holds ints (`int[]`)

---

```
array2 = [ [0, 1, 2, 3, 4],  
           [5, 6, 7, 8, 9],  
           [10, 11, 12, 13, 14],  
           [15, 16, 17, 18, 19],  
           [20, 21, 22, 23, 24] ]
```

Here is a two-dimensional array that holds ints (`int[][]`)

## 2D Arrays

We can use a **for** loop to print out each element of this array

```
array = [ 0, 1, 2, 3, 4 ]
```

```
for(int i = 0; i < array.length; i++) {  
    System.out.println(array[i]);  
}
```

Here is a one-dimensional array that holds ints (`int[]`)

```
array2 = [ [0, 1, 2, 3, 4],  
           [5, 6, 7, 8, 9],  
           [10, 11, 12, 13, 14],  
           [15, 16, 17, 18, 19],  
           [20, 21, 22, 23, 24] ]
```

We can use a nested **for** loop to print out each element of this 2D array

```
for(int i = 0; i < array2.length; i++) {  
    for(int j = 0; j < array2[i].length; j++) {  
        System.out.println(array2[i][j]);  
    }  
}
```

Here is a two-dimensional array that holds ints (`int[][]`)



# 2D Arrays

<b>x</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
<b>1</b>	1	2	3	4	5	6	7	8	9	10
<b>2</b>	2	4	6	8	10	12	14	16	18	20
<b>3</b>	3	6	9	12	15	18	21	24	27	30
<b>4</b>	4	8	12	16	20	24	28	32	36	40
<b>5</b>	5	10	15	20	25	30	35	40	45	50
<b>6</b>	6	12	18	24	30	36	42	48	54	60
<b>7</b>	7	14	21	28	35	42	49	56	63	70
<b>8</b>	8	16	24	32	40	48	56	64	72	80
<b>9</b>	9	18	27	36	45	54	63	72	81	90
<b>10</b>	10	20	30	40	50	60	70	80	90	100

Let's build a multiplication table using 2D arrays

# 2D Arrays

	0	1	2	3	4	5	6	7	8	9
<b>i</b> → 0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

## 2D Arrays

A 10x10 grid representing a 2D array. The columns are indexed 0 to 9 from left to right, with a red arrow and 'j' pointing to the first column. The rows are indexed 0 to 9 from top to bottom, with a red arrow and 'i' pointing to the first row. A code snippet in the bottom right shows nested for loops for i and j.

```
for(int i = 0; i < t; i++)  
    for(int j = 0; j < t; j++)  
        temp[i][j] = (rand() % 255);
```

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

# 2D Arrays

j  
↓

0 1 2 3 4 5 6 7 8 9

i →

0	1								
1									
2									
3									
4									
5									
6									
7									
8									
9									

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

# 2D Arrays

	0	1	2	3	4	5	6	7	8	9
0	1									
1										
2										
3										
4										
5										
6										
7										
8										
9										

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

## 2D Arrays

A 10x10 grid representing a 2D array. The columns are indexed 0 to 9 and the rows are indexed 0 to 9. A red arrow labeled **i** points to row 0, and a red arrow labeled **j** points to column 1. The cell at row 0, column 1 contains the value 2. A code snippet in the bottom right corner shows a nested loop for initializing a 2D array.

```
for(int i = 0; i < t; i++)
    for(int j = 0; j < t; j++)
        temp[i][j] = 0;
```

```
for(int i = 0; i < temp.length; i++) {
    for(int j = 0; j < temp.length; j++) {
        temp[i][j] = (i+1) * (j+1);
    }
}
```

# 2D Arrays

	0	1	2	3	4	5	6	7	8	9
0	1	2								
1										
2										
3										
4										
5										
6										
7										
8										
9										

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

# 2D Arrays

	0	1	2	3	4	5	6	7	8	9
0	1	2	3							
1										
2										
3										
4										
5										
6										
7										
8										
9										

```
for(int i = 0; i < temp.length; i++) {
    for(int j = 0; j < temp.length; j++) {
        temp[i][j] = (i+1) * (j+1);
    }
}
```



# 2D Arrays

	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

# 2D Arrays

	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1										
2										
3										
4										
5										
6										
7										
8										
9										

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

# 2D Arrays

j

↓

	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
i → 1										
2										
3										
4										
5										
6										
7										
8										
9										

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

# 2D Arrays

j

↓

	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
i → 1	2									
2										
3										
4										
5										
6										
7										
8										
9										

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

# 2D Arrays

	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1	2									
2										
3										
4										
5										
6										
7										
8										
9										

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

# 2D Arrays

**i** →

**j** ↓

	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1	2	4								
2										
3										
4										
5										
6										
7										
8										
9										

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

# 2D Arrays

	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1	2	4	6	8	10	12	14	16	18	20
2										
3										
4										
5										
6										
7										
8										
9										

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

# 2D Arrays

	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1	2	4	6	8	10	12	14	16	18	20
2	3	6	9	12	15	18	21	24	27	30
3	4	8	12	16	20	24	28	32	36	40
4	5	10	15	20	25	30	35	40	45	50
5	6	12	18	24	30	36	42	48	54	60
6	7	14	21	28	35	42	49	56	63	70
7	8	16	24	32	40	48	56			
8	9	18	27	36	45	54	63			
9	10	20	30	40	50	60	70			

```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```



# 2D Arrays

	0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9	10
1	2	4	6	8	10	12	14	16	18	20
2	3	6	9	12	15	18	21	24	27	30
3	4	8	12	16	20	24	28	32	36	40
4	5	10	15	20	25	30	35	40	45	50
5	6	12	18	24	30	36	42	48	54	60
6	7	14	21	28	35	42	49	56	63	70
7	8	16	24	32	40	48	56			
8	9	18	27	36	45	54	63			
9	10	20	30	40	50	60	70			



```
for(int i = 0; i < temp.length; i++) {  
    for(int j = 0; j < temp.length; j++) {  
        temp[i][j] = (i+1) * (j+1);  
    }  
}
```

Program 1