

# CSCI 1380

APRIL 26, 2017

# ONE- AND TWO-DIMENSIONAL ARRAYS

# DECLARATION

---

- 1D: varType varName[x]

```
int a[5];
```

- 2D: varType varName[numRows][numColumns]

```
int a[2][5];
```

# INITIALIZATION

---

- 1D: `int a[3] = { 0, 1, 2 };`
- 2D: `int a[2][3] = { {1, 5, -2}, {0, 2, 9} };`

# ASSIGNMENT

---

- 1D: `varName[index] = x;`

`a[2] = 5;`

- 2D: `varName[rowIndex][colIndex] = x;`

`a[1][2] = 5;`

# LOOPING THROUGH ARRAYS (1D)

```
int x = 5;
int a[x] = { 1, 7, -2, 0, 3 };

for (int i=0; i<x; ++i) {
    cout << a[i];
}
```

# LOOPING THROUGH ARRAYS (2D)

```
int rows = 2, cols = 3;
int a[rows][cols] = { { 1, 7, -2}, { 0, 3, 5 } };

for (int i=0; i<rows; ++i) {
    for (int j=0; j<cols; ++j) {
        cout << a[i][j];
    }
}
```

# EXERCISE

---

- Create a 2D array to store the seat reservations (row, column) of a theater with 10 rows and 10 columns.
  - Initialize the array to contain all zeroes (no reservations).
  - Reserve the following seats (update the value in a given index to 1): 1B, 1C, 5D, 8G, 9F
- Write a function that prints out the reserved seats.