CS 161
Intro to CS I

Continue 1-d Arrays, C-Strings, and Command-Line Arguments

Multidimensional Arrays

- data_type array_name[rows][cols];
  - int array[2][3];
  - int array[4][2][3];
  - int array[2][4][2][3];
- What are examples of these?
  - 2-D – Matrices, Spreadsheet, Minesweeper, Battleship, etc.
  - 3-D – Multiple Spreadsheets, (x, y, z) system
  - 4-D – (x, y, z, time) system

Initializing 2-D Arrays

- **Declaration:** int array[2][3] = {{0,0,0},{0,0,0}};
- **Individual elements:** array[0][0]=0; array[0][1]=0; array[0][2]=0; array[1][0]=0; array[1][1]=0; array[1][2]=0;
- **Loop:**

```
for(i = 0; i < 2; i++)
    for(j = 0; j < 3; j++)
        array[i][j]=0;
```

- Why do we need multiple brackets?

Reading/Printing 2-D Arrays

- Reading Array Values

```
for(i = 0; i < 2; i++)
  for(j = 0; j < 3; j++) {
    cout << “Enter a value for ” << i << “, ” << j << “: ”;
    cin >> array[i][j];
  }
```

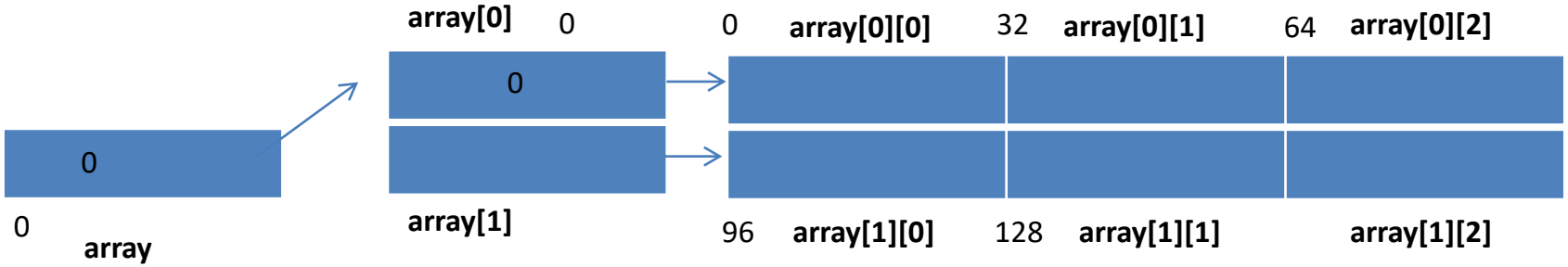
- Printing Array Values

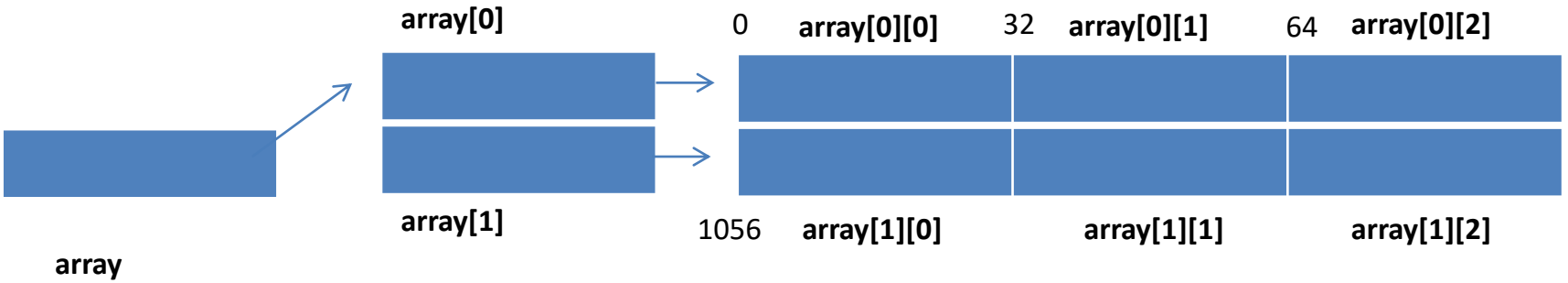
```
for(i = 0; i < 2; i++)
  for(j = 0; j < 3; j++)
    cout << “Array: ” << array[i][j] << endl;
```

Command-line Arguments

Command-line Arguments

Static vs. Dynamic 2-D arrays...





Jagged Arrays

```
int *array[2];
array[0] = new int[3];
array[1] = new int[2];
```