CS 161
Intro to CS I

2-d Arrays and Command-Line Arguments

Multidimensional Arrays

Oregon State University
College of Engineering

- 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

Oregon State University
College of Engineering

- 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];
      }
```

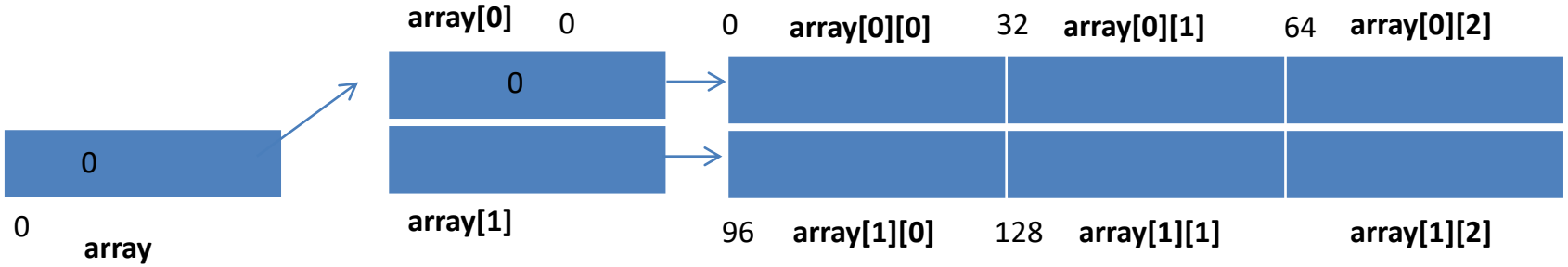
- Printing Array Values

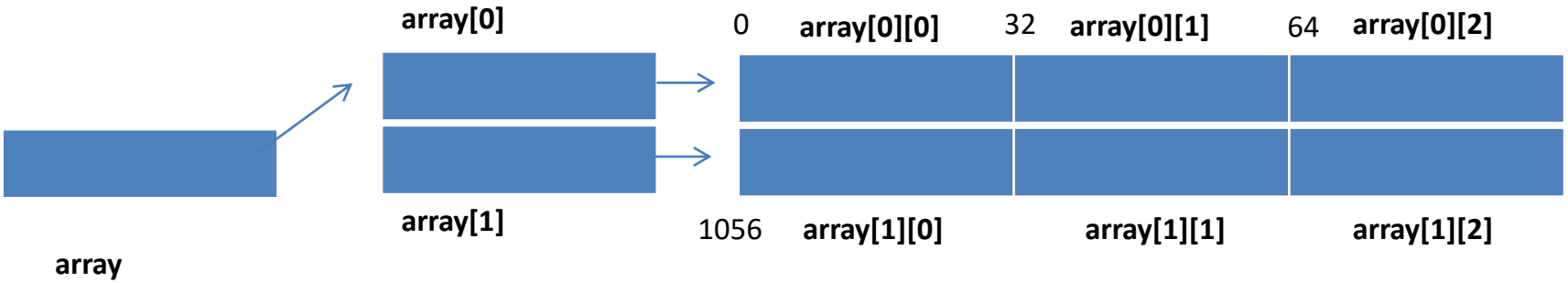
```
for(i = 0; i < 2; i++)
   for(j = 0; j < 3; j++)
         cout << "Array: " << array[i][j] <<
endl;
```

Demo...

Static vs. Dynamic 2-D arrays...

Oregon State University
College of Engineering

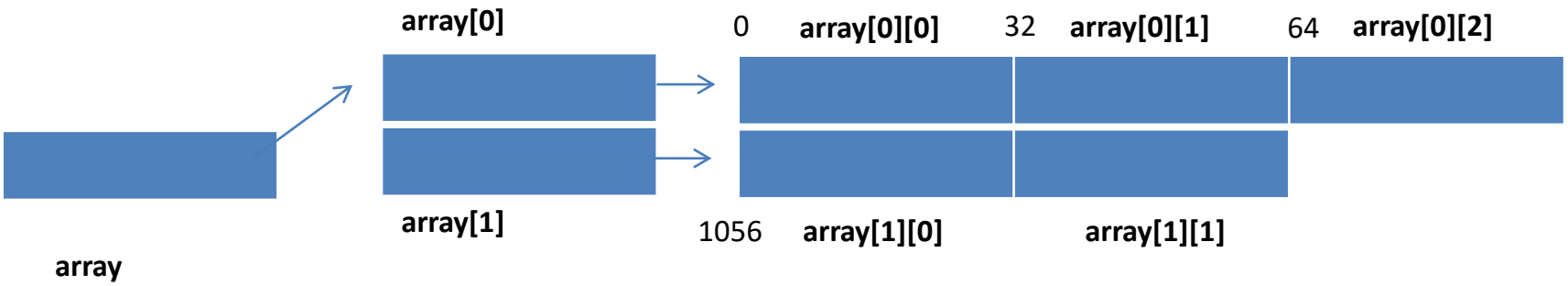




Jagged Arrays

Oregon State University
College of Engineering

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



Passing a 2-D Array (Static)

Oregon State University
College of Engineering

```
int main() {
   int array[5][5];
   …
   pass_2darray(array);
   …
}
void pass_2darray(int a[5][5]) {
   cout << "Array at zero: " << a[0][0] << endl;
}
```

OR

```
void pass_2darray(int a[][5]) {
   cout << "Array at zero: " << a[0][0] << endl;
}
```

Passing a 2-D Array (Dynamic)


```
int main() {
   int **array;
   …
   pass_2darray(array);
   …
}
void pass_2darray(int *a[]) {
    cout << "Array at zero: " << a[0][0] << endl;
}
```

OR

```
void pass_2darray(int **a) {
   cout << "Array at zero: " << a[0][0] << endl;
}
```

Passing a 2-D Array (Dynamic)

Oregon State University
College of Engineering

```
int main() {
   int *array[2];
   …
   pass_2darray(array);
   …
}
void pass_2darray(int *a[]) {
    cout << "Array at zero: " << a[0][0] << endl;
}
```

OR

```
void pass_2darray(int **a) {
   cout << "Array at zero: " << a[0][0] << endl;
}
```

Create 2-D Array in Functions

Oregon State University
College of Engineering

```
int main() {
   int **array;
   …
   array = create_2darray(rows, cols);
   …
}
int **create_2darray(int r, int c) {
    int **a;
    a = new int*[r];
    for(int i=0; i<r; i++)
        a[i] = new int[c];
    return a;

}
```

Oregon State University
College of Engineering

Create 2-D Array in Functions

```
int main() {
   int **array;
   …
create_2darray(&array, rows, cols);
   …
}
void create_2darray(int ***a, int r, int c) {
    *a = new int*[r];
    for(int i=0; i<r; i++)
        (*a)[i] = new int[c];
}
```

Create 2-D Array in Functions

Oregon State University
College of Engineering

```
int main() {
   int **array;
   …
create_2darray(array, rows, cols);
   …
}
void create_2darray(int **&a, int r, int c) {
    a = new int[r];
    for(int i=0; i<r; i++)
        a[i] = new int[c];
}
```

How does freeing memory work?

Oregon State University
College of Engineering

```
int *r[5], **s;

for(int i=0; i < 5; i++)
   r[i]=new int;
for(int i=0; i < 5; i++)
   delete r[i];

for(int i=0; i < 5; i++)
   r[i]=new int[5];
for(int i=0; i < 5; i++)
   delete [] r[i];


s=new int*[5];
for(int i=0; i < 5; i++)
   s[i]=new int[5];
for(int i=0; i < 5; i++)
   delete [] s[i];
delete [] s;
```

Command-line Arguments

Oregon State University
College of Engineering

Command-line Arguments

Oregon State University
College of Engineering

Command-line Arguments

Oregon State University
College of Engineering