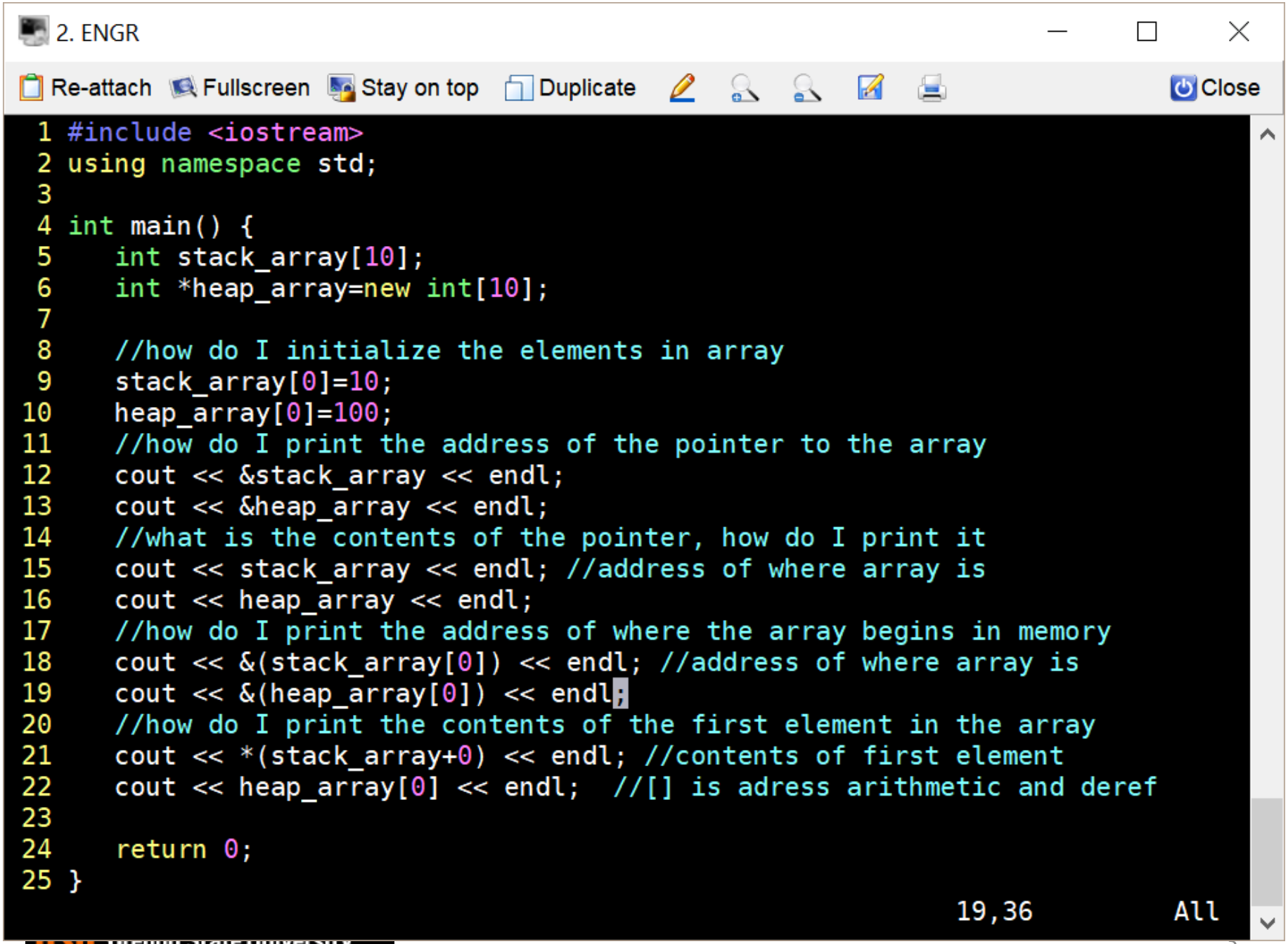
CS 161
Intro to CS I

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

Odds and Ends

- Demo Assignment 4
- Assignment 5 posted
- Veteran's Day Friday (no class/office hours)


2. ENGR
Re-attach
Fullscreen
Stay on top
Duplicate
Close
#include <iostream>
using namespace std;
int main() {
int stack_array[10];
int *heap_array=new int[10];
//how do I initialize the elements in array
stack_array[0]=10;
heap_array[0]=100;
//how do I print the address of the pointer to the array
cout << &stack_array << endl;
cout << &heap_array << endl;
//what is the contents of the pointer, how do I print it
cout << stack_array << endl; //address of where array is
cout << heap_array << endl;
//how do I print the address of where the array begins in memory
cout << &(stack_array[0]) << endl; //address of where array is
cout << &(heap_array[0]) << endl;
//how do I print the contents of the first element in the array
cout << *(stack_array+0) << endl; //contents of first element
cout << heap_array[0] << endl; //[] is adress arithmetic and deref
return 0;
}
19,36
All

Passing a 1-D Array (Static/Dynamic)

```
int main() {
   int array[5];
   …
   pass_1darray(array);
   …
}
void pass_1darray(int *a) {
   cout << “Array at zero: ” << a[0] << endl;
}
```

OR

```
void pass_1darray(int a[]) {
   cout << “Array at zero: ” << a[0] << endl;
}
```

Class Exercise

- How would I do the above in a function?
- How would I create an array in a function?

What are C-style strings?

- Ended by ‘\0’ character
- Need to include <cstring>

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];
  }
```

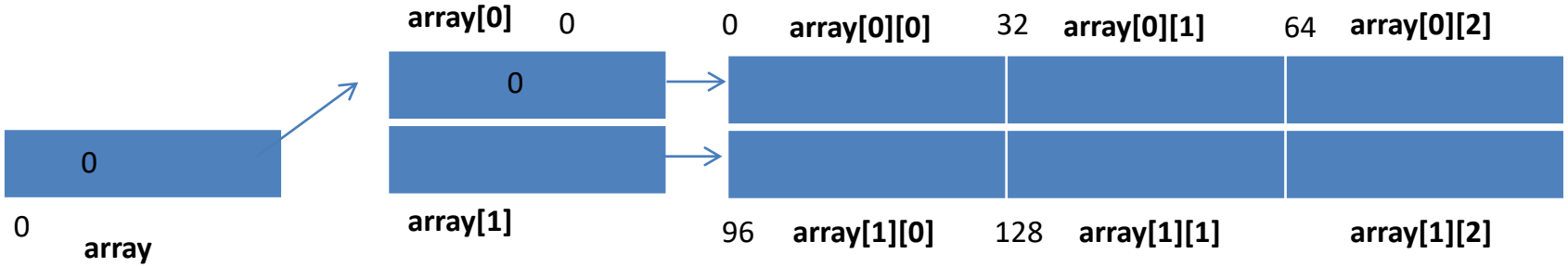
- Printing Array Values

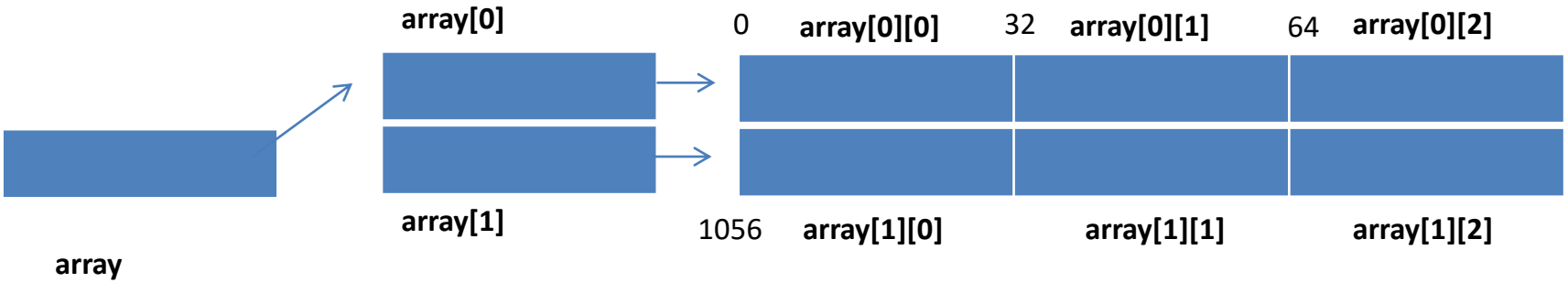
```
for(i = 0; i < 2; i++)
  for(j = 0; j < 3; j++)
    cout << “Array: ” << array[i][j] << endl;
```

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];
```