CS 161
Intro to CS I

Command-Line Arguments

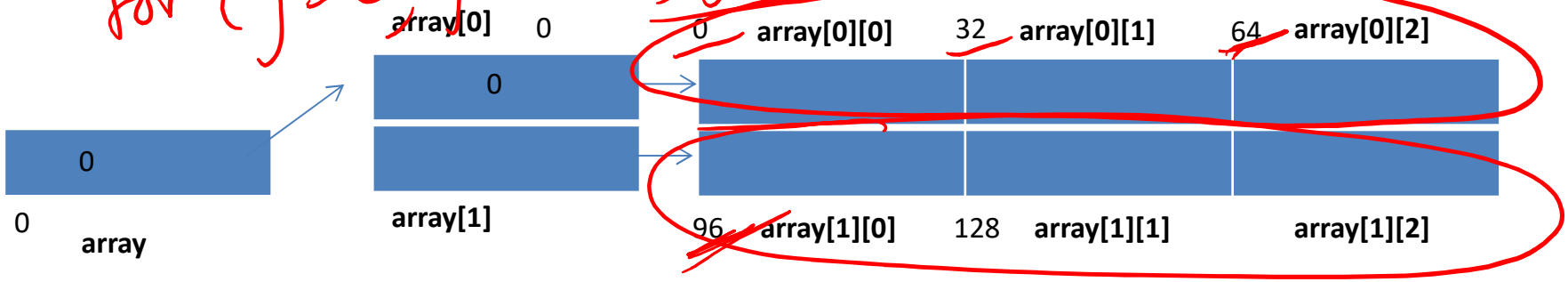
Odds and Ends

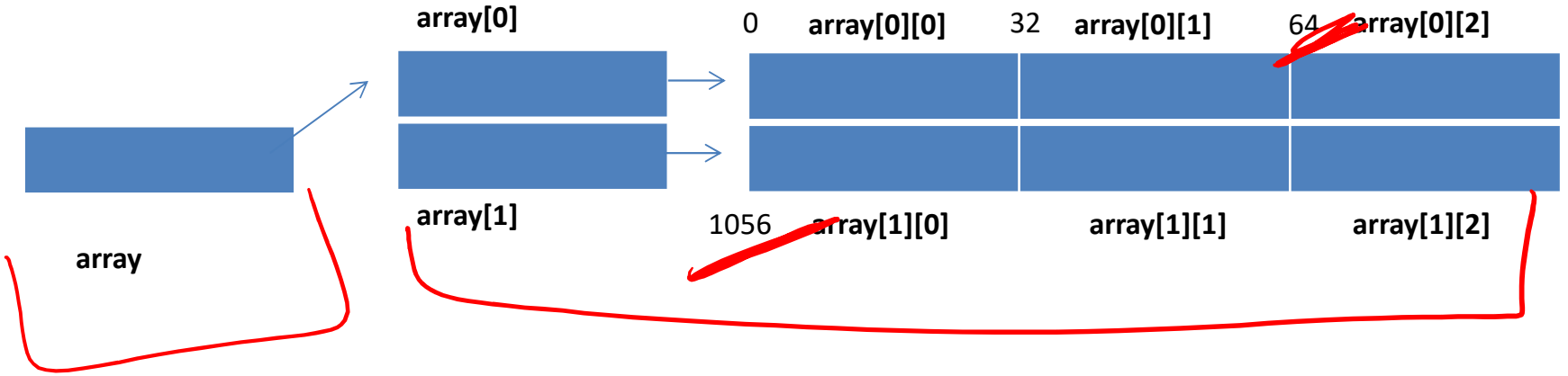
Oregon State University
College of Engineering

- Last week to demo Assignment 5
- Grades are accurate
- Final Exam: Thursday (3/21) in WNGR 151
  - Section 001, 12-1pm
  - Section 002, 1-2pm
  - If you don't take final, then I'll average Exam 1 & 2

- Questions?

Static vs. Dynamic 2-D arrays...

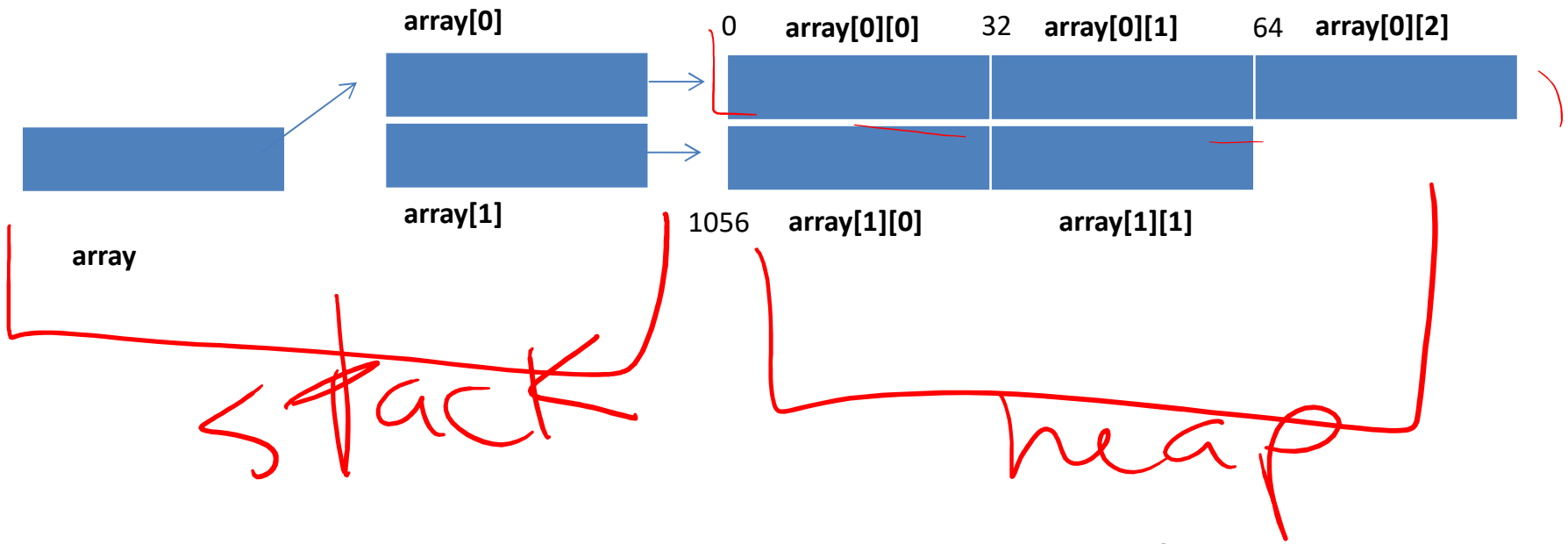




Jagged Arrays

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;
}
```

stride convention

Passing a 2-D Array (Dynamic)

Oregon State University
College of Engineering

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

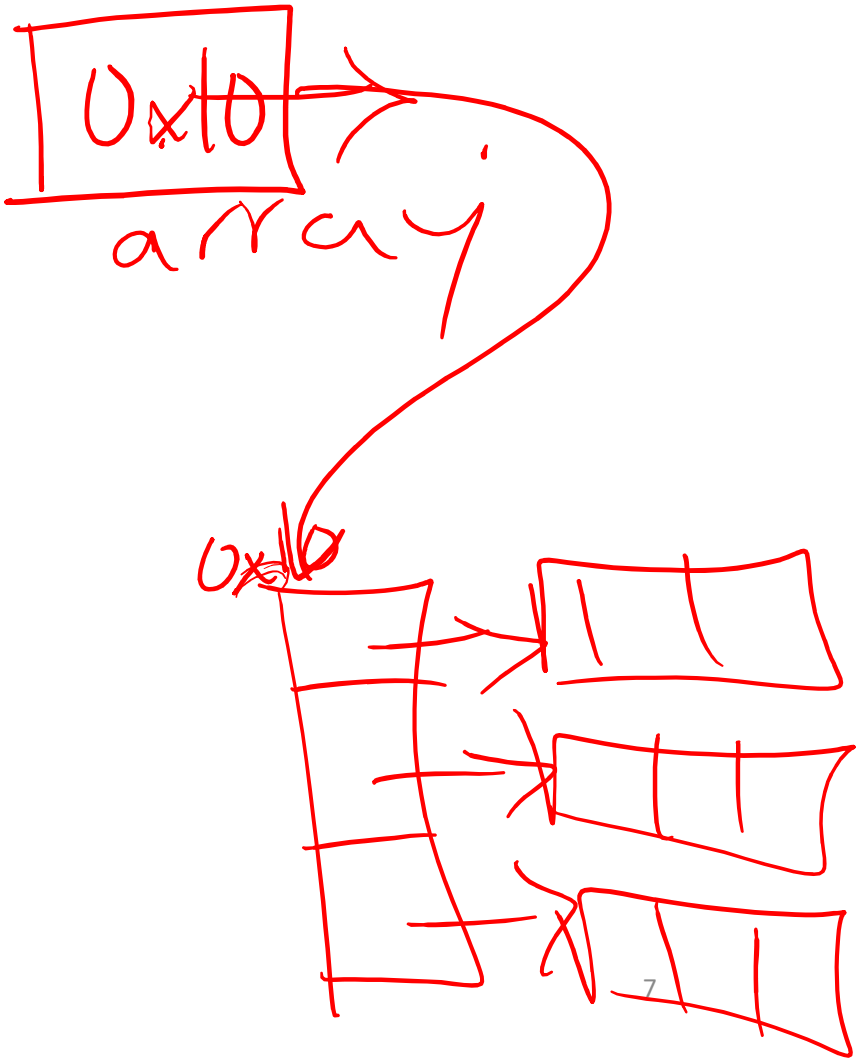
OR

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

Create 2-D Array in Functions

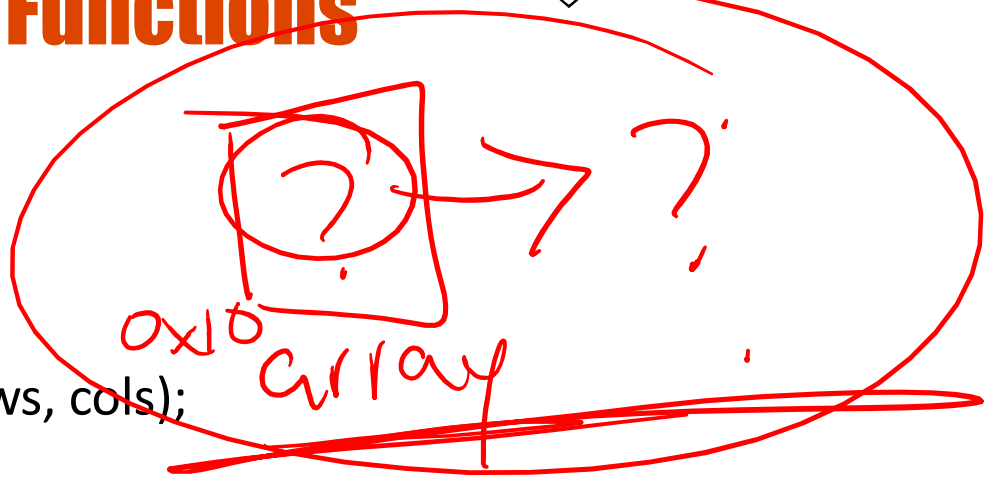
Oregon State University
College of Engineering

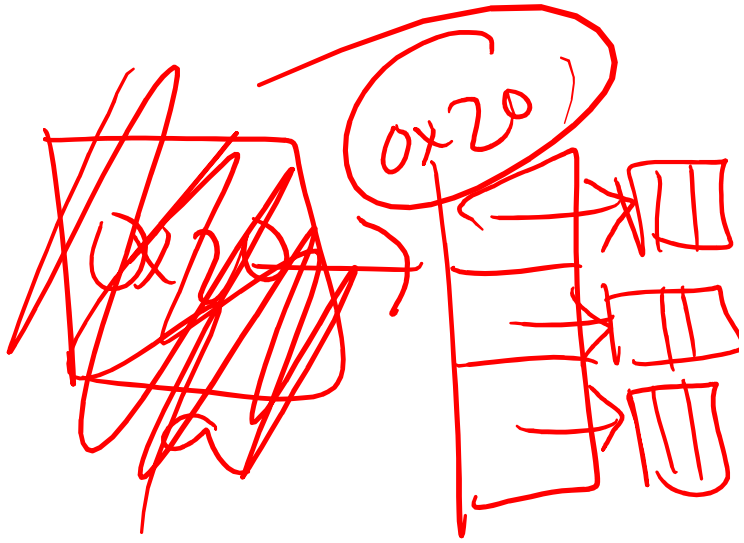
```
int main() {
  int **array;
  …
  array = create_2darray(rows, cols);
  …
}
int **create_2darray(int r, int c) {
   int **a;
   a = new int*[r];
   for(int i=0; i<r; i++)
      a[i] = new int[c];
   return a;
}
```



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];
}
```

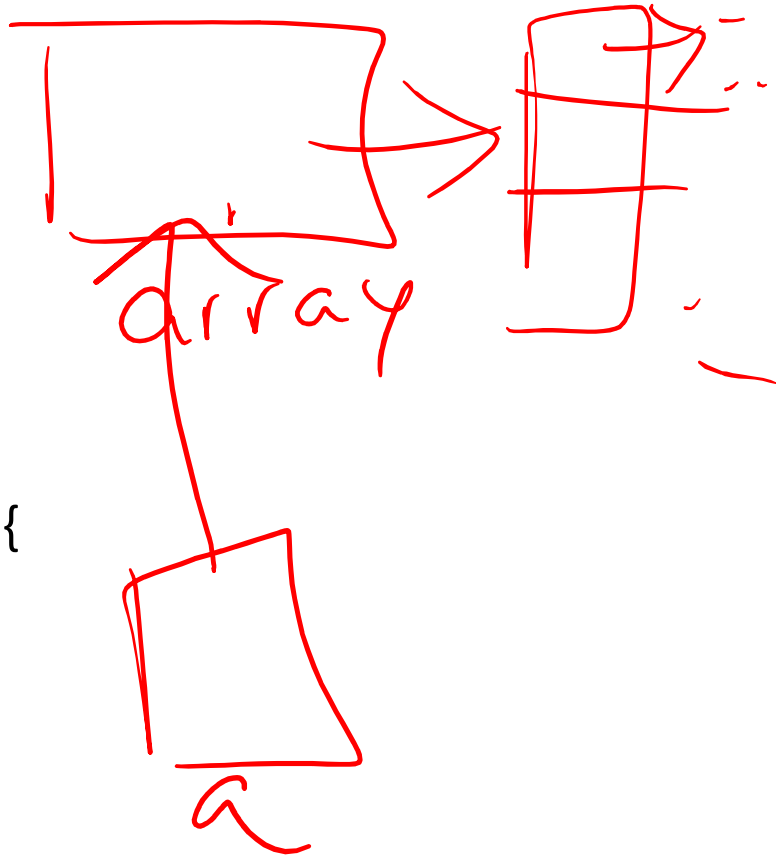




Create 2-D Array in Functions

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;
```

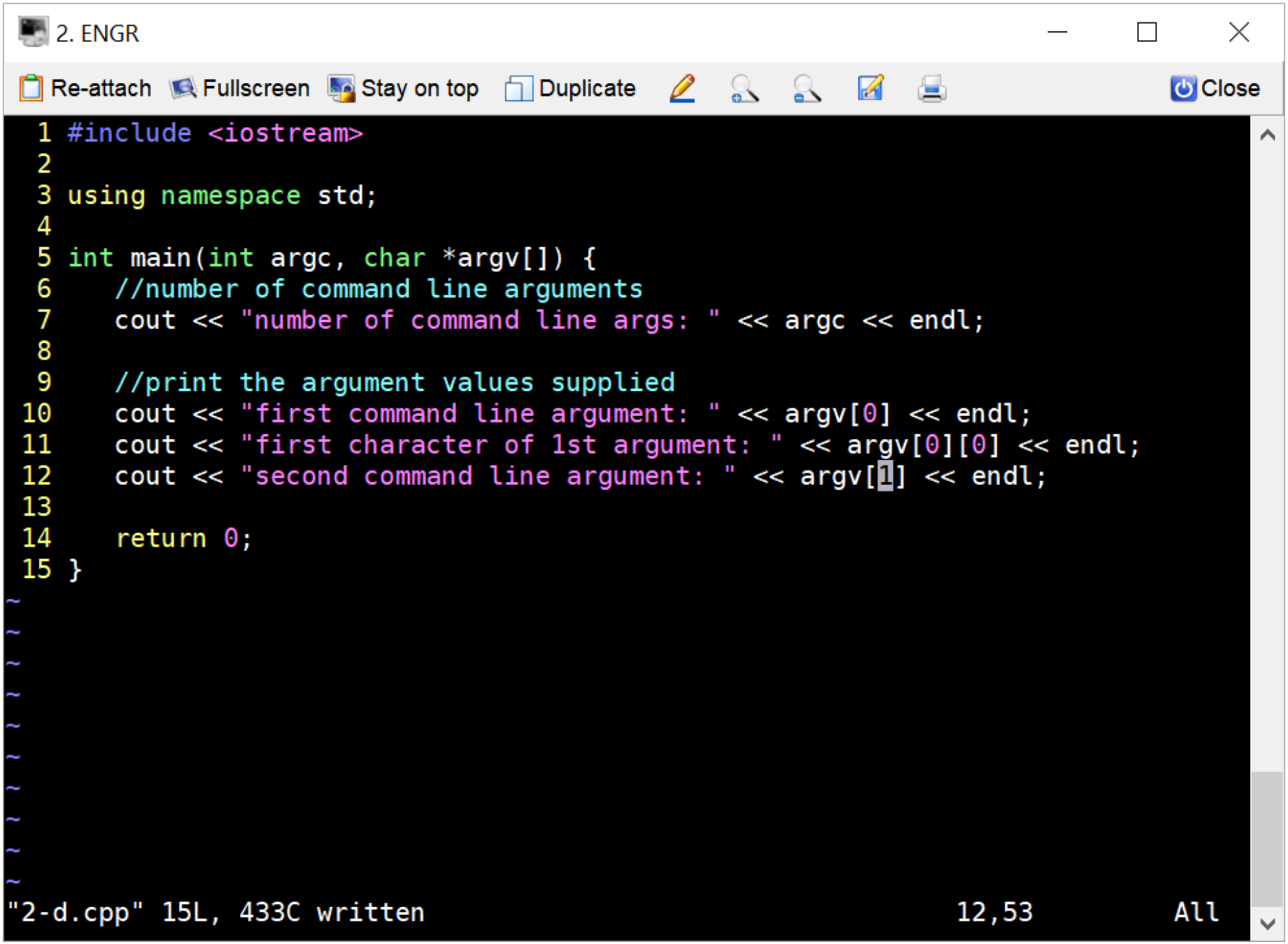
Command-line Arguments

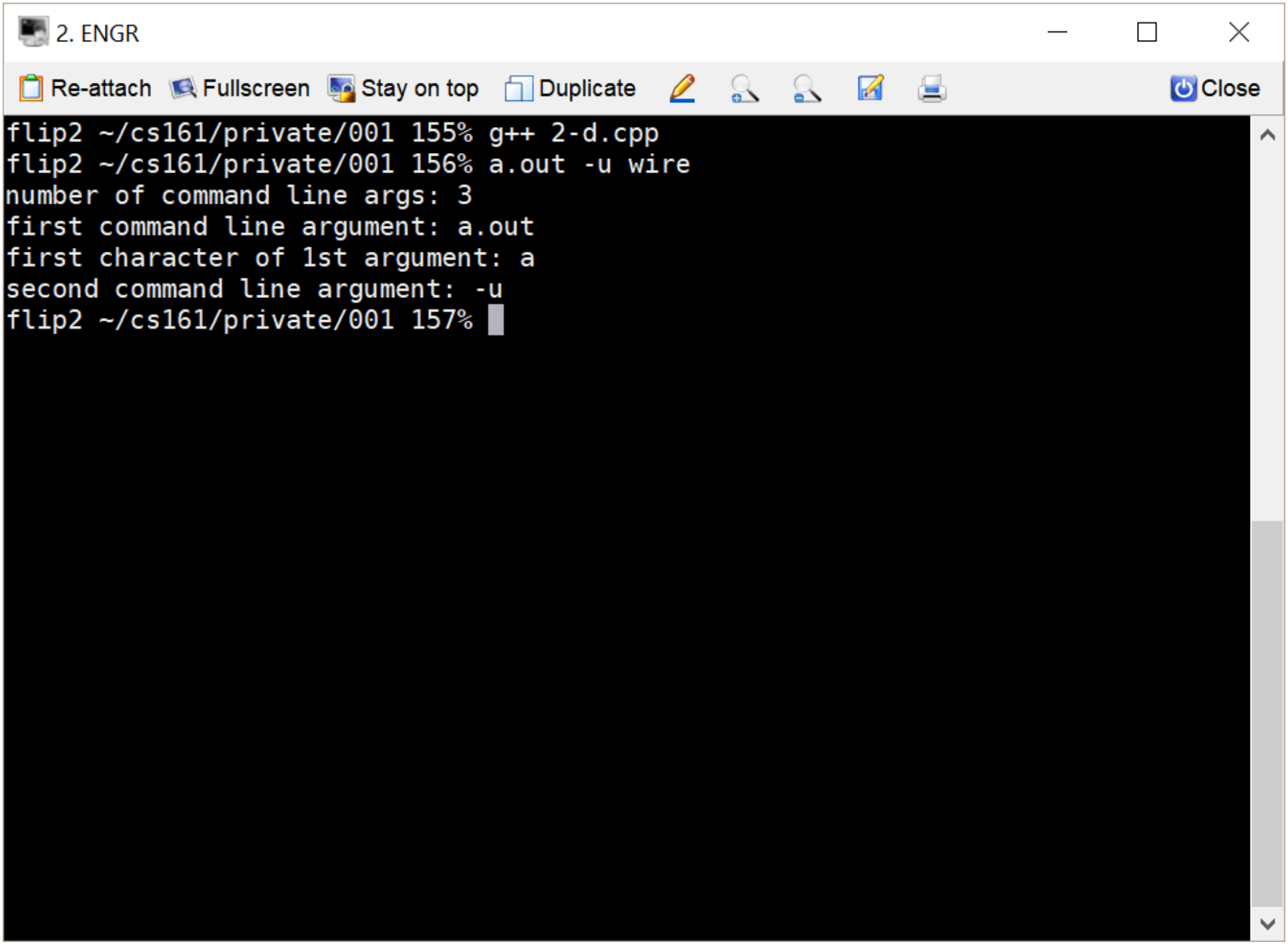
Oregon State University
College of Engineering

- Supplied at runtime

```
int main(int argc, char *argv[]) {


   return 0;
}
```


2. ENGR
Re-attach
Fullscreen
Stay on top
Duplicate
Close
1 #include <iostream>
2
3 using namespace std;
4
5 int main(int argc, char *argv[]) {
6 //number of command line arguments
7 cout << "number of command line args: " << argc << endl;
8
9 //print the argument values supplied
10 cout << "first command line argument: " << argv[0] << endl;
11 cout << "first character of 1st argument: " << argv[0][0] << endl;
12 cout << "second command line argument: " << argv[1] << endl;
13
14 return 0;
15 }
"2-d.cpp" 15L, 433C written
12,53
All


2. ENGR
Re-attach
Fullscreen
Stay on top
Duplicate
Close
flip2 ~/cs161/private/001 155% g++ 2-d.cpp
flip2 ~/cs161/private/001 156% a.out -u wire
number of command line args: 3
first command line argument: a.out
first character of 1st argument: a
second command line argument: -u
flip2 ~/cs161/private/001 157%