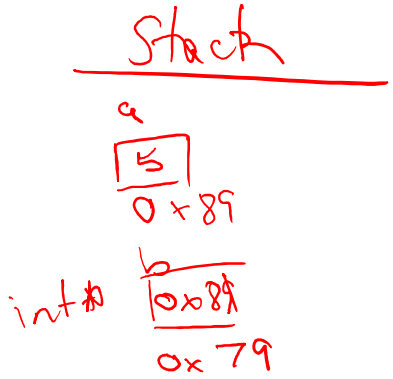
CS 162, Lecture 2: Pointer, Array and Struct Review

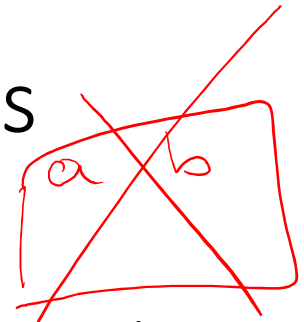
6 April 2018

Pointers



- Pointers == memory addresses
- Variable declaration: int a = 5;
  - Creates a variable on the stack of size int with the value 5.
- Pointer declaration: int* b = &a;
  - Creates a pointer variable on the stack which can hold an address of an int and sets the value of the pointer (the address the pointer points to) to the address of a
- Dereferencing Pointer: cout << *b << endl;
  - Will print the value stored at the address which b points to

Pointers and Functions



- void func(int a, int b);
  - Prototype indicating that void func is expecting two parameters of type int to be passed by value
  - Recall pass by value copies the value being passed into the formal parameters which are scoped to this function, any changes made to the values in the function will not reflect outside this scope
- void func(int* a, int b);
  - Prototype indicating that void func is expecting two parameters, one of type int* (an address of an int) and one of type int.
  - Since a is of type int* (an address) it will have to be dereferenced throughout the function if the value stored at that address is to be used. a may also receive a new address in the function. Changes made to this variable will persist beyond the scope of this function

Demo Pointers

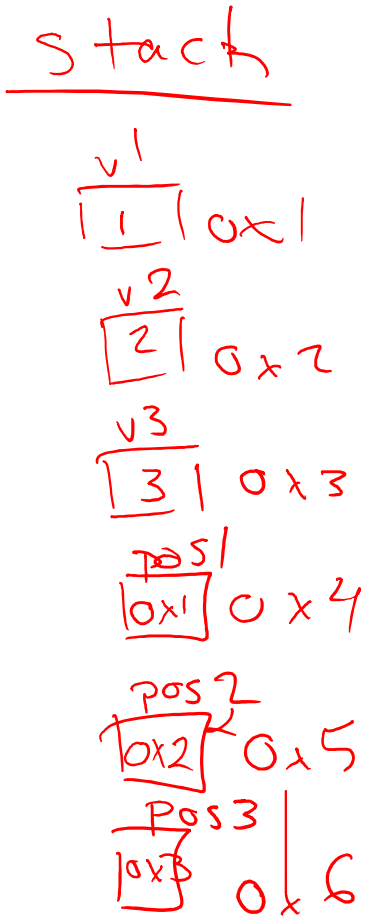
```
int main() {
        int v1 = 1, v2 = 2, v3 = 3;
        int *pos1, *pos2, *pos3;

        pos1 = &v1;
        pos2 = &v2;
        pos3 = &v3;

        print(pos1, pos2, pos3);

        for (int i = 0; i < 4; i++) {
                rotate(&pos1, &pos2, &pos3);
                print(pos1, pos2, pos3);
        }

        return 0;
}
```



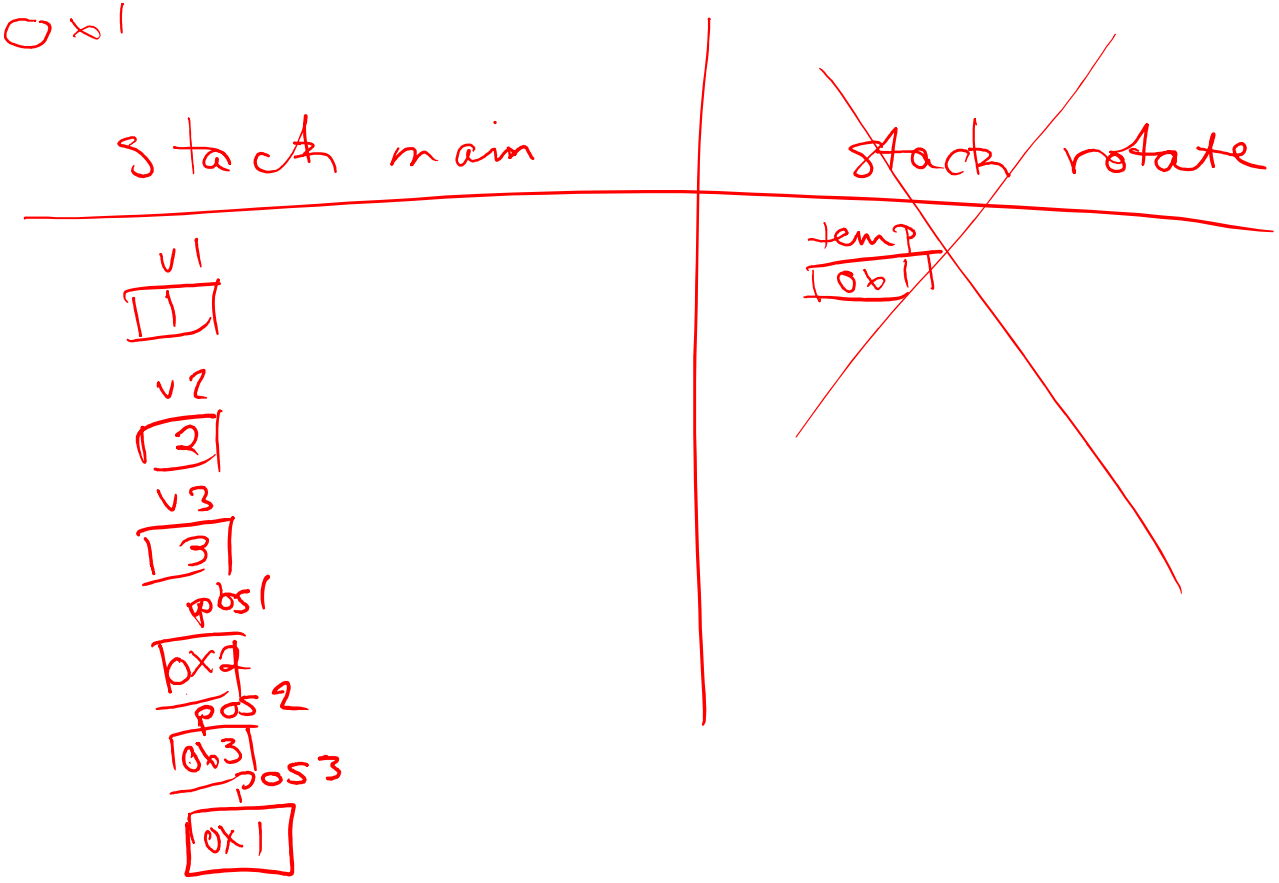
```
void rotate(int** pos1, int** pos2, int** pos3){
        int *temp = *pos1;
        *pos1 = *pos2;
        *pos2 = *pos3;
        *pos3 = temp;
}
```

rotate(&pos1, &pos2, &pos3)

*pos1 = *pos2; 0x2

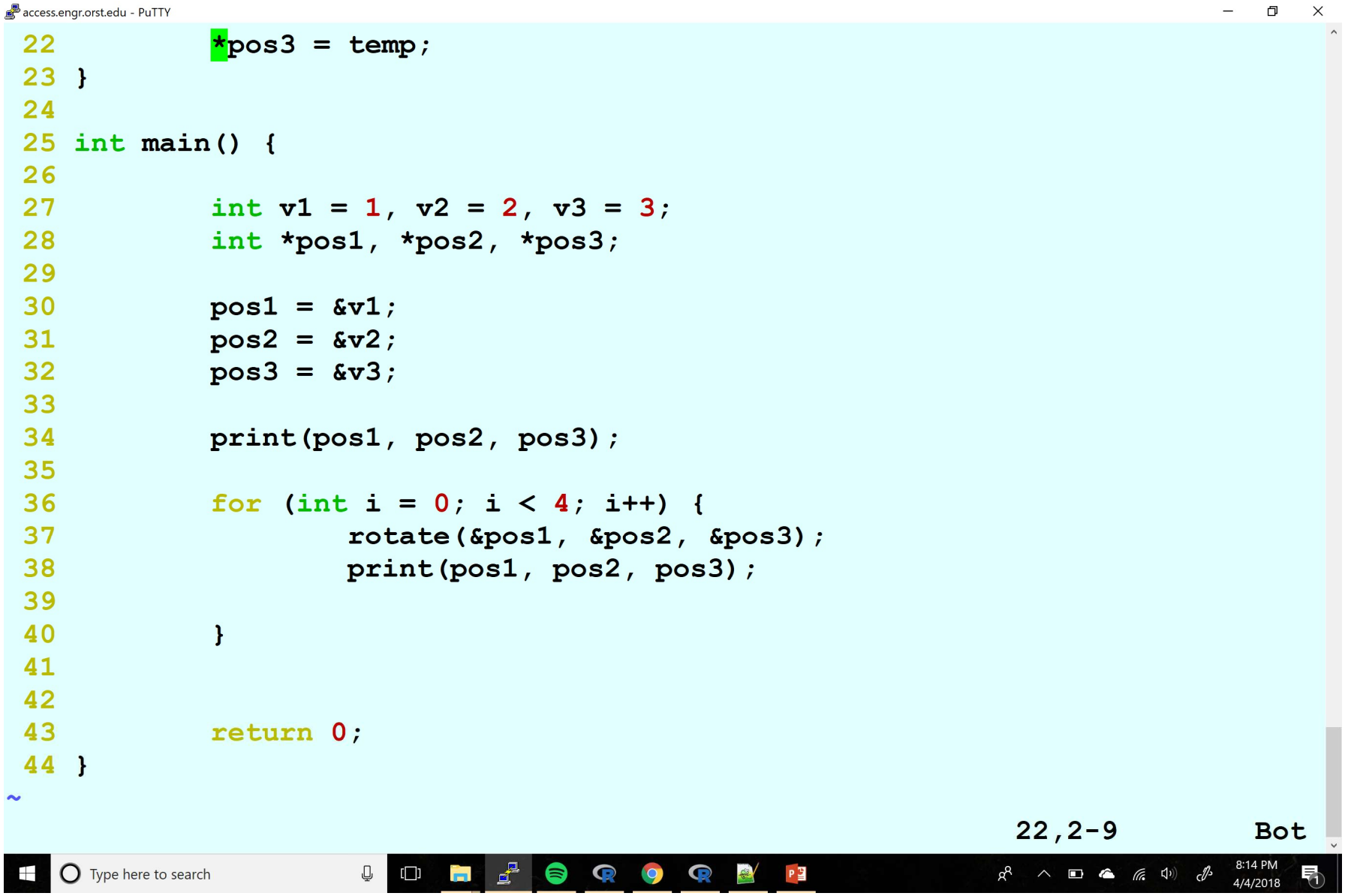
*pos2 = *pos3; 0x3

*pos3 = temp; 0x1



access.engr.orst.edu - PuTTY

```
 1 #include <iostream>
 2
 3 using namespace std;
 4
 5 void print(int* pos1, int* pos2, int* pos3){
 6         cout << "Addresses Stored in the Pointers" << endl;
 7         cout << "pos1: " << pos1 << endl;
 8         cout << "pos2: " << pos2 << endl;
 9         cout << "pos3: " << pos3 << endl;
10         cout << "Values Stored at the Addresses Stored in the Pointers" <
   < endl;
11         cout << "value at pos1: " << *pos1 << endl;
12         cout << "value at pos2: " << *pos2 << endl;
13         cout << "value at pos3: " << *pos3 << endl;
14         cout << endl;
15
16 }
17
18 void rotate(int** pos1, int** pos2, int** pos3){
19         int *temp = *pos1;
20         *pos1 = *pos2;
21         *pos2 = *pos3;
22         *pos3 = temp;
23 }
"pointers.cpp" 44L, 842C                                19,2-9        Top
```


access.engr.orst.edu - PuTTY
22 *pos3 = temp;
23 }
24
25 int main() {
26
27 int v1 = 1, v2 = 2, v3 = 3;
28 int *pos1, *pos2, *pos3;
29
30 pos1 = &v1;
31 pos2 = &v2;
32 pos3 = &v3;
33
34 print(pos1, pos2, pos3);
35
36 for (int i = 0; i < 4; i++) {
37 rotate(&pos1, &pos2, &pos3);
38 print(pos1, pos2, pos3);
39
40 }
41
42
43 return 0;
44 }
~
22,2-9 Bot
Type here to search
8:14 PM
4/4/2018

Arrays

- An array is of one data type and its memory is stored contiguously
- Static Arrays: created on the stack and are of a fixed size
- Dynamic Arrays: created on the heap and their size may change during runtime
- Arrays may be of one or more dimensions

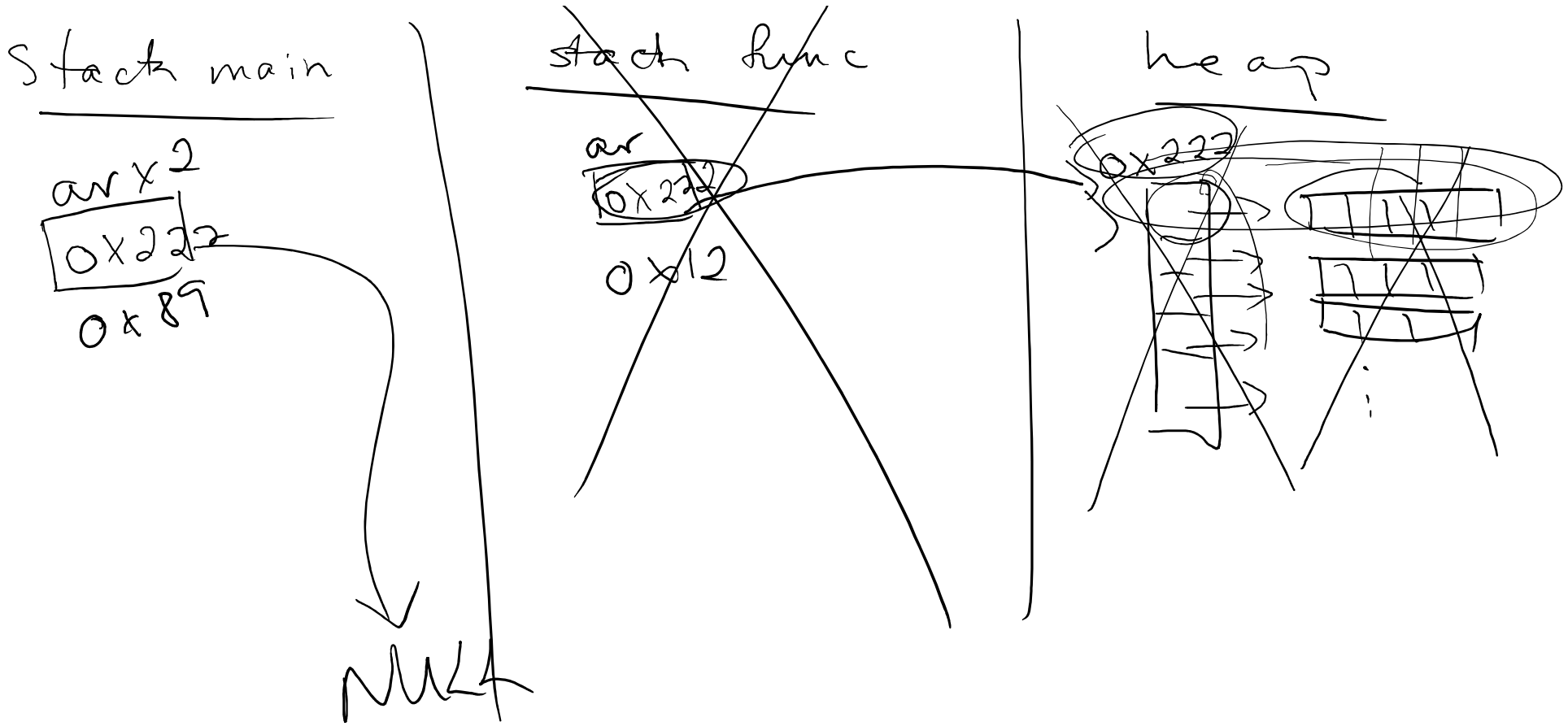
Array Demo

```
void print(int* ar, int len) {
        for (int i = 0; i < len; i++) {
            cout << ar[i] << " ";
        }
        cout << endl;
        cout << endl;
}
void rotate(int* ar, int len) {
        int temp = ar[0];
        for (int i = 0; i < len-1; i++) {
                ar[i] = ar[i+1];
        }
        ar[len-1] = temp;
}
int main() {
        int ar[5] = {1,2,3,4,5};
        print(ar, 5);
        for (int i = 0; i < 4; i++) {
                rotate(ar, 5);
                print(ar, 5);
        }
        return 0;
}
```

ar = 0x11

ar [0]

```
void pop_arx2(int ar[][5], int r, int c) {
	for(int i = 0; i < r; i++) {
		for(int j = 0; j < c; j++) {
			ar[i][j] = i*j;
		}
	}
}
void print_arx2(int ar[][5], int r, int c) {
	for(int i = 0; i < r; i++) {
		for(int j = 0; j < c; j++) {
			cout << ar[i][j] << " ";
		}
		cout << endl;
	}
	cout << endl;
}
int main() {
	int arx2[5][5];
	pop_arx2(arx2, 5, 5);
	print_arx2(arx2, 5, 5);
	return 0;
}
```


Stack main
arr x2
0x222
0x89
NULL
stack func
arr
0x222
0x12
heap
0x222

access.engr.orst.edu - PuTTY

```
 1 #include <iostream>
 2
 3 using namespace std;
 4
 5 void print(int* ar, int len) {
 6         for (int i = 0; i < len; i++) {
 7                 cout << ar[i] << " ";
 8         }
 9         cout << endl;
10         cout << endl;
11 }
12
13 void rotate(int* ar, int len) {
14         int temp = ar[0];
15         for (int i = 0; i < len-1; i++) {
16                 ar[i] = ar[i+1];
17         }
18         ar[len-1] = temp;
19 }
20
21 void pop_arx2(int ar[][5], int r, int c) {
22         for(int i = 0; i < r; i++) {
23                 for(int j = 0; j < c; j++) {
24                         ar[i][j] = i*j;
"static_arrays.cpp" 52L, 817C                                 21,1          Top
```

access.engr.orst.edu - PuTTY

```
19 }
20
21 void pop_arx2(int ar[][5], int r, int c) {
22         for(int i = 0; i < r; i++) {
23                 for(int j = 0; j < c; j++) {
24                         ar[i][j] = i*j;
25                 }
26         }
27 }
28
29 void print_arx2(int ar[][5], int r, int c) {
30         for(int i = 0; i < r; i++) {
31                 for(int j = 0; j < c; j++) {
32                         cout << ar[i][j] << " ";
33                 }
34                 cout << endl;
35         }
36         cout << endl;
37 }
38
39 int main() {
40
41         int ar[5] = {1,2,3,4,5};
42         print(ar, 5);
```

21,1 64%

Type here to search

8:15 PM
4/4/2018

access.engr.orst.edu - PuTTY

```
29 void print_arx2(int ar[][5], int r, int c) {
30         for(int i = 0; i < r; i++) {
31                 for(int j = 0; j < c; j++) {
32                         cout << ar[i][j] << " ";
33                 }
34                 cout << endl;
35         }
36         cout << endl;
37 }
38
39 int main() {
40
41         int ar[5] = {1,2,3,4,5};
42         print(ar, 5);
43         for (int i = 0; i < 4; i++) {
44                 rotate(ar, 5);
45                 print(ar, 5);
46         }
47         int arx2[5][5];
48         pop_arx2(arx2, 5, 5);
49         print_arx2(arx2, 5, 5);
50
51         return 0;
52 }
```

52,1 Bot

access.engr.orst.edu - PuTTY

```
 1 #include <iostream>
 2
 3 using namespace std;
 4
 5 void init_pop_arx1(int** ar, int len) {
 6         (*ar) = new int[len];
 7         for (int i = 0; i < len; i++) {
 8                 (*ar)[i] = i;
 9         }
10 }
11
12 void rotate(int* ar, int len) {
13         int temp = ar[0];
14         for (int i = 0; i < len-1; i++) {
15                 ar[i] = ar[i+1];
16         }
17         ar[len-1] = temp;
18 }
19
20
21
22 void print_arx1(int* ar, int len) {
23         for (int i = 0; i < len; i++) {
24                 cout << ar[i] << " ";
```

24,2-16 Top

Type here to search

8:17 PM
4/4/2018

access.engr.orst.edu - PuTTY

```
22 void print_arx1(int* ar, int len) {
23         for (int i = 0; i < len; i++) {
24                 cout << ar[i] << " ";
25         }
26         cout << endl;
27         cout << endl;
28 }
29
30
31 int** init_pop_arx2(int r, int c) {
32         int** ar = new int*[r];
33         for (int i = 0; i < r; i++) {
34                 ar[i] = new int[c];
35                 for (int j = 0; j < c; j++) {
36                         ar[i][j] = i*j;
37                 }
38         }
39         return ar;
40 }
41
42 void print_arx2(int** ar, int r, int c) {
43         for (int i = 0; i < r; i++) {
44                 for (int j = 0; j < c; j++) {
45                         cout << ar[i][j] << " ";
```

24,2-16 38%

Type here to search

8:17 PM
4/4/2018

access.engr.orst.edu - PuTTY

```
40 }
41
42 void print_arx2(int** ar, int r, int c) {
43         for (int i = 0; i < r; i++) {
44                 for (int j = 0; j < c; j++) {
45                         cout << ar[i][j] << " ";
46                 }
47                 cout << endl;
48         }
49         cout << endl;
50 }
51
52 void delete_arx2(int** ar, int r) {
53         for (int i = 0; i < r; i++)
54                 delete [] ar[i];
55         delete  [] ar;
56         //ar = NULL;
57 }
58
59 int main() {
60
61         int* ar;
62         int r = 5;
63         init_pop_arx1(&ar, r);
```

43,2-9 70%

access.engr.orst.edu - PuTTY

```
58
59 int main() {
60
61         int* ar;
62         int r = 5;
63         init_pop_arx1(&ar, r);
64         print_arx1(ar, r);
65         for (int i = 0; i < r; i++) {
66                 rotate(ar, r);
67                 print_arx1(ar, r);
68         }
69         delete [] ar;
70         ar = NULL;
71
72         int c = 5;
73         int** arx2 = init_pop_arx2(r, c);
74         print_arx2(arx2, r, c);
75         delete_arx2(arx2, r);
76         arx2 = NULL;
77
78         return 0;
79 }
~
~
                                                    58,0-1          Bot
```

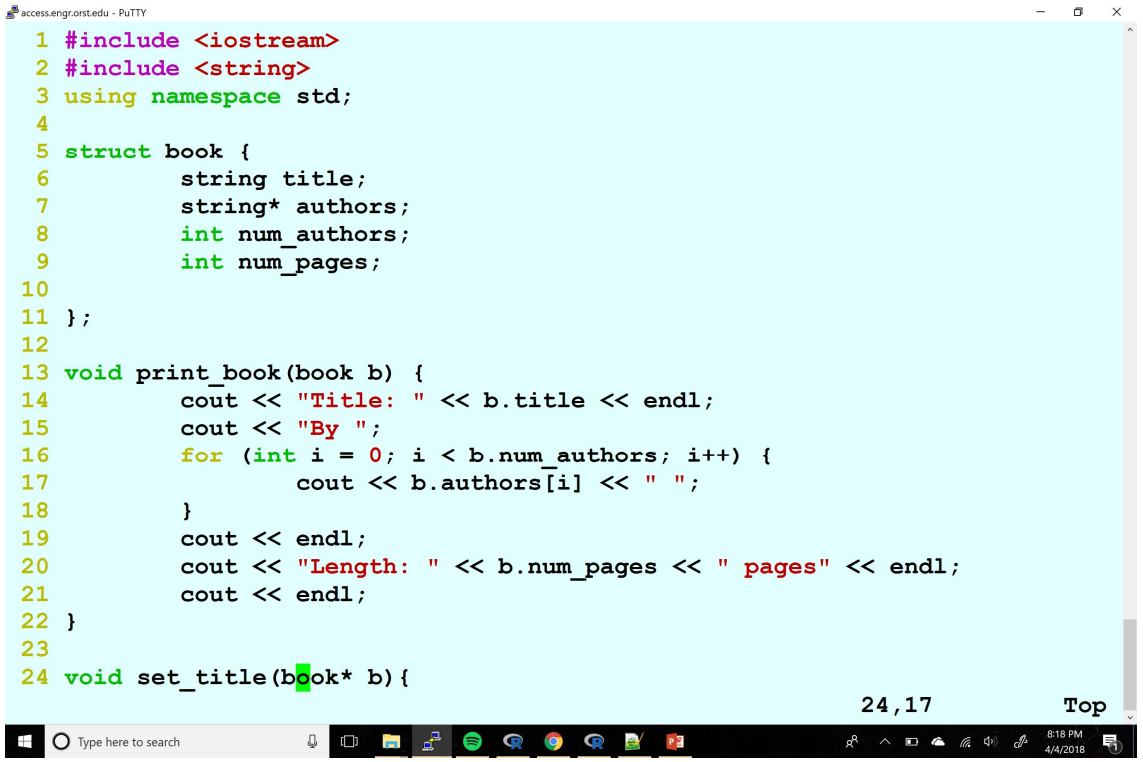
Structs

- User defined objects
- Container which holds many variables of different types

```
struct my_obj {
        string name;
        int some_stat;
        float* some_list_of_stats;
};
```

- Can be used as any other data type with some extra features
  - Access members using the dot operator
    - Ex: my_obj.name = “Obj1”;
  - If the struct needs to be dereferenced use the arrow
    - Ex: my_obj->name = “Obj1”; (*my_obj).name = “Obj1”;

Struct Demo


access.engr.orst.edu - PuTTY
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 struct book {
6 string title;
7 string* authors;
8 int num_authors;
9 int num_pages;
10
11 };
12
13 void print_book(book b) {
14 cout << "Title: " << b.title << endl;
15 cout << "By ";
16 for (int i = 0; i < b.num_authors; i++) {
17 cout << b.authors[i] << " ";
18 }
19 cout << endl;
20 cout << "Length: " << b.num_pages << " pages" << endl;
21 cout << endl;
22 }
23
24 void set_title(book* b){
24,17 Top
Type here to search
8:18 PM
4/4/2018

access.engr.orst.edu - PuTTY

```
22 }
23
24 void set_title(book* b){
25         cout << "What is the title? ";
26         getline(cin, b->title);
27 }
28
29 void set_authors(book* b) {
30         cout << "How many authors? ";
31         cin >> b->num_authors;
32         cin.ignore();
33         cin.clear();
34         b->authors = new string[b->num_authors];
35         for (int i = 0; i < b->num_authors; i++) {
36                 cout << "Who is author " << i+1 << "? ";
37                 getline(cin, b->authors[i]);
38         }
39 }
40
41 void set_num_pages(book* b) {
42         cout << "How many pages? ";
43         cin >> b->num_pages;
44         cin.ignore();
45         cin.clear();
```

25,10-17 61%

Type here to search

8:18 PM
4/4/2018

access.engr.orst.edu - PuTTY

```
40
41 void set_num_pages(book* b) {
42         cout << "How many pages? ";
43         cin >> b->num_pages;
44         cin.ignore();
45         cin.clear();
46 }
47
48
49
50 int main() {
51
52         book b;
53         //How should we call the functions?
54
55         print_book(b);
56
57         return 0;
58 }
~
~
~
~
~
                                                    40,0-1          Bot
```

Type here to search

8:18 PM
4/4/2018