

CS 161

Intro to CS I

1-d Arrays (Static vs. Dynamic)

Odds and Ends

- Last day to demo Assignment 3
- Assignment 4 due Sunday



z top

```
if (n == 2)
```

```
    print(n/2, col)
```

```
else
```

```
    pattern(n/2, col)
```

```
    print(n, col)
```

```
if (n == 2)
```

```
    print(n/2, col + n)
```

```
else
```

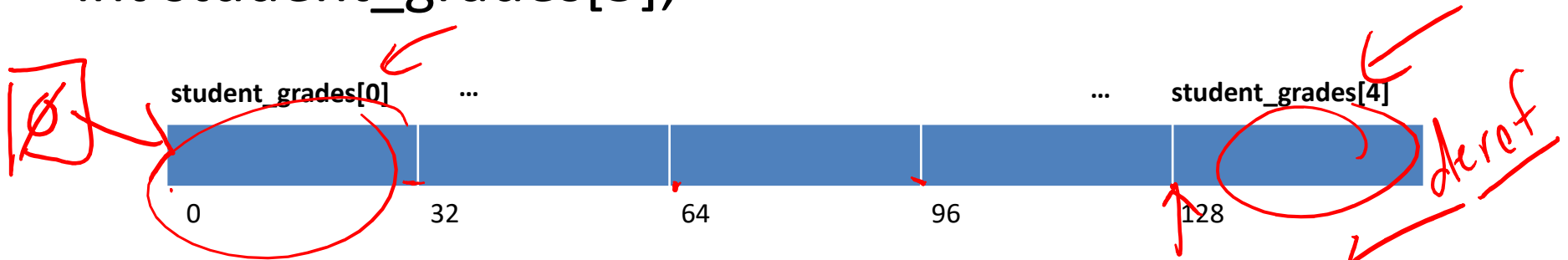
```
    pattern(n/2, col + n)
```

= middle

= bottom

Create 1-D Array

```
int student_grades[5];
```



- How do you access each item? *name (ele index)*
- What does the array name represent? *address of where array begins*
- Why is the array name the address of 1st element?
- What are the initial values? *garbage*

Initialize/Assign Values

- **Declaration**

```
int student_grades[5] = {0, 0, 0, 0, 0};
```

new array
{0, 0, 0, 0, 0}?

at the time of declaration

- **Individual Elements**

```
student_grades[0]=0;
```

...

```
student_grades[4]=0;
```

for loop

- **Why is this incorrect?**

```
student_grades={0, 0, 0, 0, 0};
```

address

~~BAD~~

Initialize/Assign Values...

- **Using a Loop**

- While Loop Example:**

```
i=0;  
while (i<5) {  
    student_grades[i]=0;  
    i++;  
}
```

- For Loop Example:**

```
for(i=0; i<5; i++)  
    student_grades[i]=0;
```

*do not
equal the
length*

- Which is better to use with arrays and why?

Read/Print 1-D Array Values

- Read Values From User

```
for(i=0; i<5; i++) {  
    cout << "Enter final grade for student: ";  
    cin >> student_grades[i];  
}
```

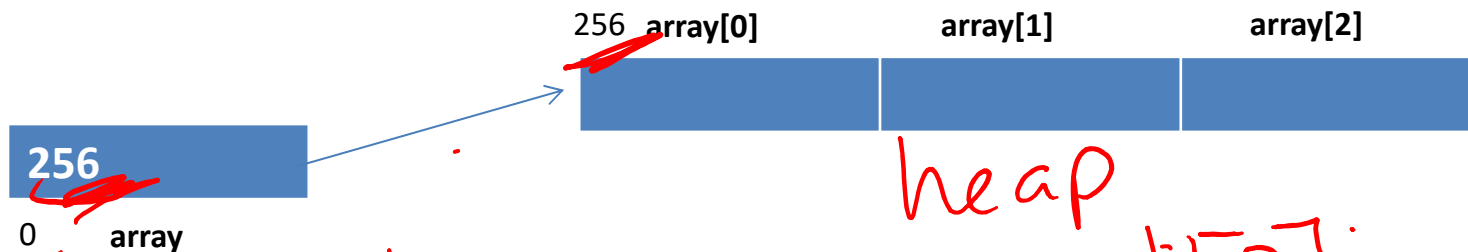
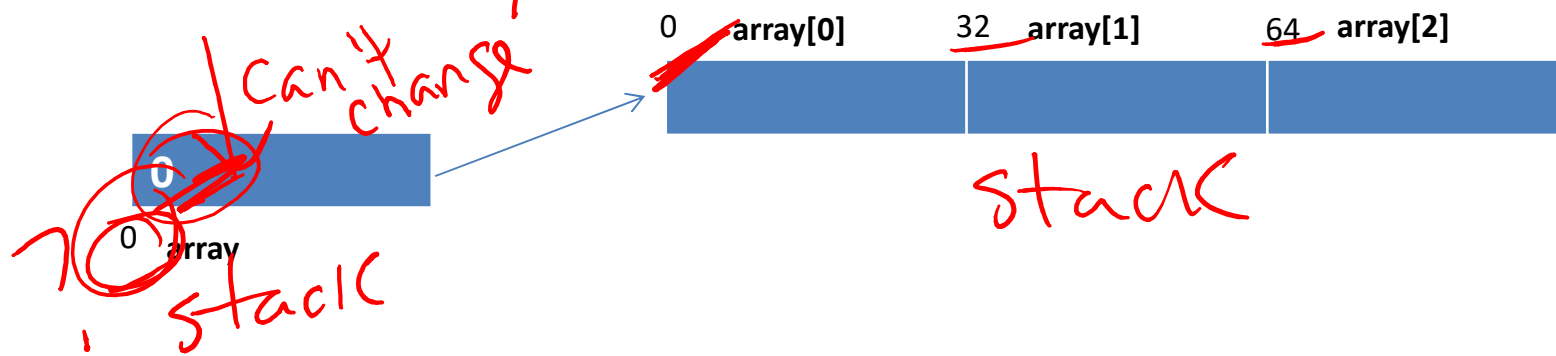
- Print Values

```
for (i=0; i<5; i++) {  
    cout << "Student\'s final grade is " << student_grades[i] << endl;  
}
```

Static vs. Dynamic 1-D arrays...

`int array[3];`

constant, self-referential arrays



heap
`int *array = new int[3];`
256

How does creating and freeing memory work?

```
int *p, *q;
```

```
p=new int;
```

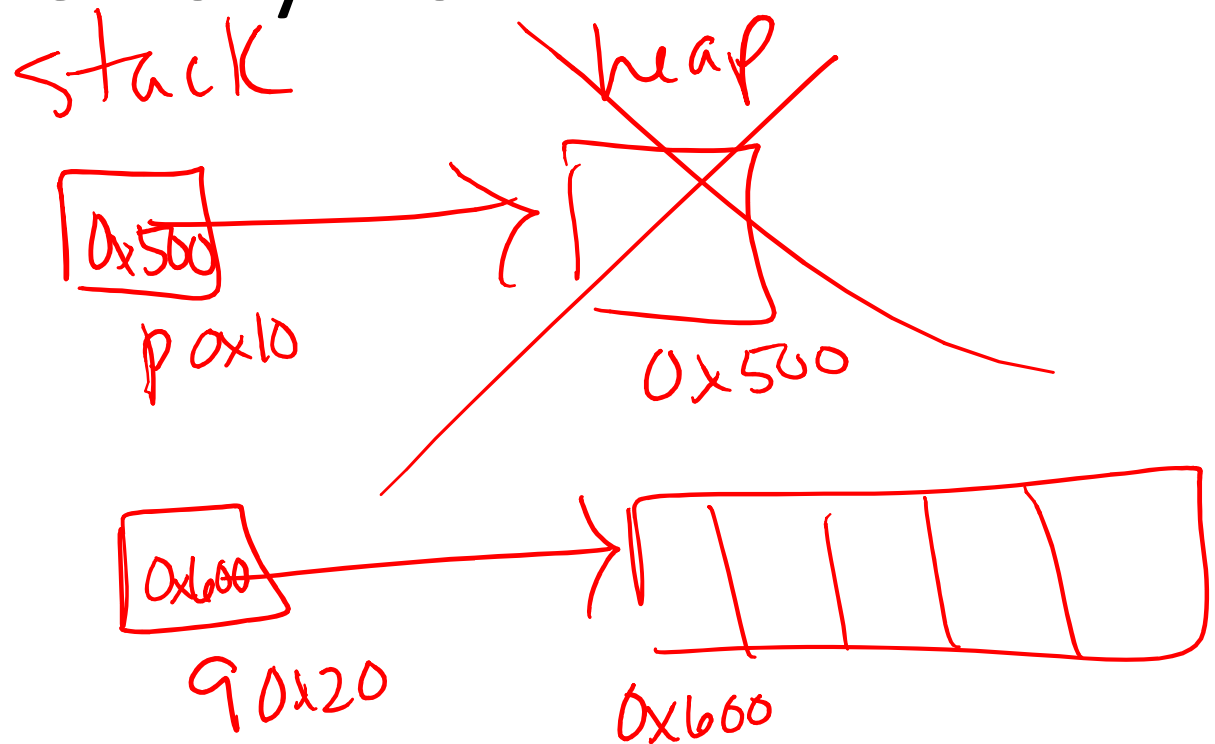
```
q=new int[5];
```

```
delete p;
```

```
delete [] q;
```

*the mem
p points to*

*the arrays
q points to*




```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int stack_array[10];
7     int heap_array=new int[10];
8
9     //how do I initialize the elements in array
10
11
12     //how do I print the address of the pointer to the array
13
14
15     //what is the contents of the pointer, how do I print it
16
17
18     //how do I print the address of where the array begins in memory
19
20
21     //how do I print the contents of the first element in the array
22
23
24     return 0;
25 }
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int stack_array[10];
6     int *heap_array=new int[10];
7
8     //how do I initialize the elements in array
9     stack_array[0]=10;
10    heap_array[0]=100;
11    //how do I print the address of the pointer to the array
12    cout << &stack_array << endl;
13    cout << &heap_array << endl;
14    //what is the contents of the pointer, how do I print it
15    cout << stack_array << endl; //address of where array is
16    cout << heap_array << endl;
17    //how do I print the address of where the array begins in memory
18    cout << &(stack_array[0]) << endl; //address of where array is
19    cout << &(heap_array[0]) << endl;
20    //how do I print the contents of the first element in the array
21    cout << *(stack_array+0) << endl; //contents of first element
22    cout << heap_array[0] << endl; //[ ] is adress arithmetic and deref
23
24    return 0;
25 }
```