

# **CS 161**

## **Intro to CS I**

Finish Recursion/Begin Memory Model



## Odds and Ends

- Assignment 4 demo
- Peer Reviews Thursday

• Extra credit added to A5  
- add<sub>n</sub> to the zip file  
journal extra

man zip  
unzip

man tar

Dyna

```
2. ENGR
Re-attach Fullscreen Stay on top Duplicate
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     char *p; //create the pointer
7     p=new char; //now make it point to new character on heap
8
9     cout << "p lives: " << &p << endl;
10    cout << "contents of p: " << (void *) p << endl;
11    cout << "contents of heap: " << *p << endl;
12    *p='j';
13    cout << "contents of heap: " << *p << endl;
14    delete p; //delete what p points to
15
16    //while(1) {
17        char j='k';
18        p=&j; //can make p point to character on stack, but don't delete it
19        cout << "contents of p: " << (void *) p << endl;
20        cout << "contents of stack: " << *p << endl;
21
22        p=new char; //make p point to character on heap, make sure to delete
23        cout << "contents of p: " << (void *) p << endl;
24        cout << "contents of heap: " << *p << endl;
25        *p='e';
26        cout << "contents of heap: " << *p << endl;
27        delete p;
28    //}
29    return 0;
-- INSERT --
```

22,7

Top

University  
engineering

```
2. ENGR
Re-attach Fullscreen Stay on top Duplicate
flip3 ~/cs161/private/001 194% g++ arrays.cpp
flip3 ~/cs161/private/001 195% a.out
p lives: 0x7fffba55c868
contents of p: 0x106d010
contents of heap:
contents of heap: j
contents of p: 0x7fffba55c867
contents of stack: k
contents of p: 0x106d010
contents of heap:
contents of heap: e
flip3 ~/cs161/private/001 196% valgrind a.out
```

University  
engineering



# What is an Array?

- **Array (ar-ray)  $n$ .** An ordered arrangement of related items.
  - Example: Array of colors in a rainbow.
    - Related items?
    - Ordered arrangement?
  - Class examples?
  - Computer Science
    - Same data type/data structure
    - Contiguous memory locations

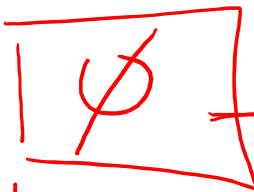
# Create 1-D Array



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

stack

# items

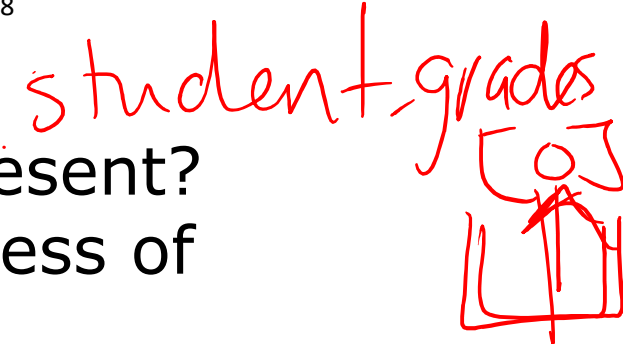


student\_grades

How do you access each item?

- What does the array name represent?
- Why is the array name the address of 1<sup>st</sup> element?
- What are the initial values?

pointer





# Initialize/Assign Values

- Declaration

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

*{13} initialize*

- Individual Elements

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

...

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

*student\_grades[i]*

- Why is this incorrect?

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

*\* (student\_grades + 1)*



# 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;
```

- 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

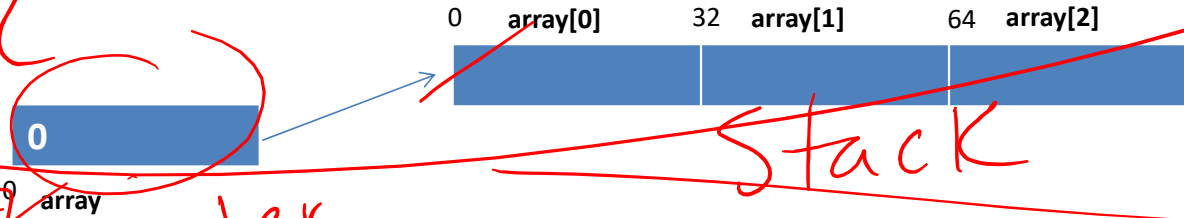
```
for (i=0; i<5; i++) {  
    cout << "Student\'s final grade is " << student_grades[i] << endl;  
}
```

# Static vs. Dynamic 1-D arrays...



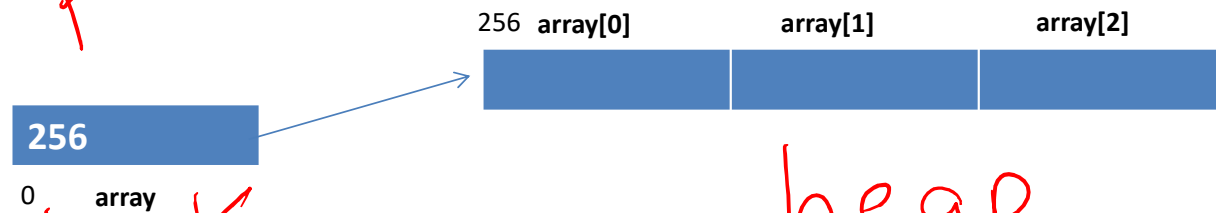
*int array [3]; array = 0*

*stack*



*Stack*

*constant self-ref pointer.*



*stack*

*heap*

*int \*array = new int [3];  
delete [] array;*

```
2. ENGR
Re-attach Fullscreen Stay on top Duplicate
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     char *p; //create the pointer
7     p=new char[10]; //now make it point to new character array on heap
8
9     for(int i=0; i<10; i++)
10         p[i]='j';
11
12     cout << "p lives: " << &p << endl;
13     cout << "contents of p: " << (void *) p << endl;
14     cout << "contents of heap: " << *p << endl; //contents of p[0]
15     cout << "contents of p[1]: " << *p << endl; //contents of 2nd element
16     *p='k';
17     cout << "contents of heap: " << *p << endl;
18     delete [] p; //delete what p points to
19
20     //while(1) {
21         p=new char; //make p point to character on heap, make sure to delete
22         cout << "contents of p: " << (void *) p << endl;
23         cout << "contents of heap: " << *p << endl;
24         *p='e';
25         cout << "contents of heap: " << *p << endl;
26         delete p;
27     //}
28     return 0;
29 }
```

-- INSERT -- 29,2 All