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 to the zip file
  - journal extra

man, zip
unzip
man far
```cpp
#include <iostream>

using namespace std;

int main() {
  char *p; //create the pointer
  p=new char; //now make it point to new character on heap
  cout << "p lives: " << &p << endl;
  cout << "contents of p: " << (void *) p << endl;
  cout << "contents of heap: " << *p << endl;
  *p='j';
  cout << "contents of heap: " << *p << endl;
  delete p; //delete what p points to

  //while(1) {
    char j='k';
    p=&j; //can make p point to character on stack, but don't delete it
    cout << "contents of p: " << (void *) p << endl;
    cout << "contents of stack: " << *p << endl;
  } //}

  p=new char; //make p point to character on heap, make sure to delete
  cout << "contents of p: " << (void *) p << endl;
  cout << "contents of heap: " << *p << endl;
  *p='e';
  cout << "contents of heap: " << *p << endl;
  delete p;

  return 0;
}  
```
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
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
int student_grades[5];

- How do you access each item?
  - What does the array name represent?
  - Why is the array name the address of 1st element?
  - What are the initial values?
Initialize/Assign Values

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

• Individual Elements
  
  ```c
  student_grades[0]=0;  // Initialize
  ...                
  student_grades[4]=0;
  ```

• Why is this incorrect?
  
  ```c
  student_grades={0, 0, 0, 0, 0};  // Error
  ```
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...

```cpp
int array[3], array = 0;
```

Constant self-ref pointer.

```
0 array
```

```
```

Stack

heap

```
int *array = new int[3];
delete [] array;
```
```cpp
#include <iostream>

using namespace std;

int main() {
    char *p; //create the pointer
    p = new char[10]; //now make it point to new character array on heap

    for(int i = 0; i < 10; i++)
        p[i] = 'i';

    cout << "p lives: " << &p << endl;
    cout << "contents of p: " << (void *)p << endl;
    cout << "contents of heap: " << *p << endl; //contents of p[0]
    cout << "contents of p[1]: " << *p << endl; //contents of 2nd element
    *p = 'k';
    cout << "contents of heap: " << *p << endl;
    delete [] p; //delete what p points to

    //while(1) {
    p = new char; //make p point to character on heap, make sure to delete
    cout << "contents of p: " << (void *)p << endl;
    cout << "contents of heap: " << *p << endl;
    *p = 'e';
    cout << "contents of heap: " << *p << endl;
    delete p;
    //}
return 0;
}
```