

# Today - Lecture 11 - CS162

- 1) Introduction to Pointers
- 2) Dynamically Allocated Arrays
- 3) Using Dynamically Allocated Arrays in our show list program
  - creating arrays sized just right at run time for names
  - creating the array of items sized just right at run time
  - deallocating (releasing) that memory

Announcements:

# Pointers

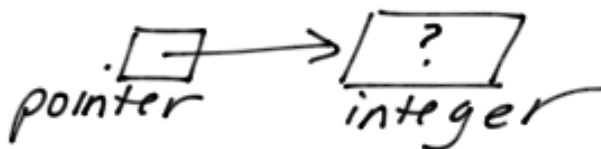
- 1) A pointer "Variable" holds a memory address
- 2) It is best to set pointers to NULL if they are not pointing anywhere
- 3) NULL is just a #define constant for  $\emptyset$

```
int * pointer = NULL;
```



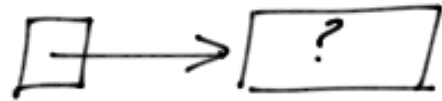
- 4) use "new" to allocate memory "dynamically" as the program is running

```
pointer = new int;
```



# Dynamically Allocated Memory

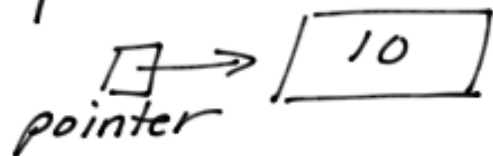
```
int * pointer = new int;
```



1) To Access The Memory we must get there indirectly through the pointer variable

This is called **DEREFERENCING**

```
* pointer = 10;
```



```
cout << * pointer ;  
          10
```

# Deallocation of memory

## General

1) Memory for local variables (including pointer variables) is automatically released at the end of the block `{ }` in which the variable was defined

2) `However` memory allocated with `NEW` is not automatically released until you use `delete`

3) For every use of `new` there should be a corresponding use of `delete`

```
int * ptr = new int;
```

```
delete ptr; // releases the memory  
           // that ptr points to.
```

```
} ←
```

# Dynamically Allocated Arrays

```
int length;
cout << "How many scores are there?";
cin >>length; cin.ignore(100,'\n');

float * scores;
float grade = 0.0;

scores = new float[length]; //dynamically allocated array

cout <<"Please enter " <<length <<" scores: ";
for (int i=0; i<length; ++i)
{
    cin >>scores[i];
    cin.ignore(100,'\n');
}

//calculate the average score
for (int i=0; i<length; ++i)
{
    grade+= scores[i];
}
grade /= length;

//we are done...
delete [] scores;
```

# Pointer Arithmetic

$$\text{array}[i] == *(\text{array} + i)$$

Starting  
address

index is the  
offset

temp  
address

Go There!

---

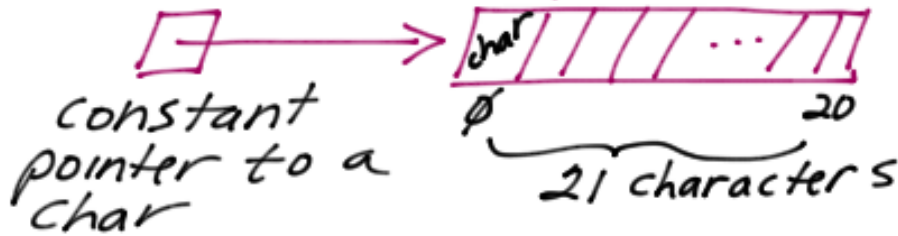
$$*(i + \text{array})$$

$i[\text{array}]$

# Interesting...

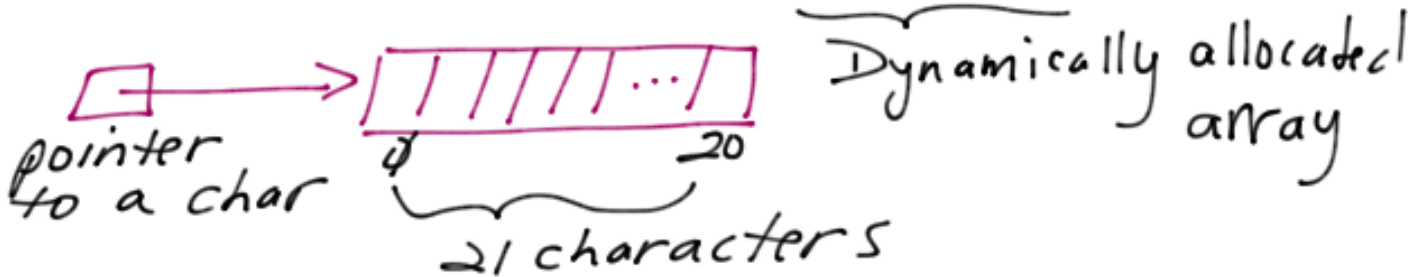
a pointer can point to one item OR  
the first of many items contiguously  
arranged.

`char name [21];` ← called a  
Statically Allocated  
Array



versus

`char * name = new char [21];` ← this can be  
a variable!



`delete [] name ;` // once we are done with  
// the array

# Dynamically Allocated Array of characters - sized just right! -

```
char temp[100];  
char * name;  
  
cout <<"Please enter your full name: ";  
cin.get(temp,100,'\n');  
cin.ignore(100,'\n');  
  
name = new char[strlen(temp) + 1];  
strcpy(name,temp);  
  
cout <<"You entered " <<name <<endl;  
  
//when done  
delete [] name;
```



*Examine These*

*(not all are correct)*

```
char temp[100];  
char * name;
```

```
cout <<"Please enter your full name: ";  
cin.get(temp,100,'\n');  
cin.ignore(100,'\n');
```

```
name = new char[strlen(temp) + 1];
```

```
name = new char[strlen(temp+1)];
```

```
name = new char[strlen(temp)] +1;
```