

## Eng 101 – Day 6 1/19/04

Arrays and function review

## Admin

- P0 due tonight.
- HW1 returned Monday
  - Will try to get things back in a week from now on. Getting graders up to speed.
- HW1 and inlab1 answers posted.
  - Be sure you are watching the webpage once or twice a week.

## Near Future

- Friday we introduce “structures”
- Monday & Friday of next week we will introduce very little in the way of language constructs
  - Instead we will focus on algorithms, programming and a more formal description of C++
    - We’ve been “learning by example.” Time to learn the rules.
- On Wednesday of next week we will tackle multi-dimensional arrays.

## More future

- After that we have
  - Two major language constructs to consider:
    - File I/O and Strings
  - “How a computer works”
    - Just what is an int? A double?
    - How does a computer execute code?
    - What is computer memory?
  - Some minor language constructs
    - For loops and a few other minor things.

# Today

- Arrays
  - Start again (Friday was a while ago!) with a simple example
  - Write the sorting code.
- Functions and Arrays
  - Using arrays as arguments to a function
  - Receiving those arguments as parameters.

```
include<iostream>
using namespace std;

main()
{
    int a[5]={1,3,5,7,9};
    int b;

    a[3]=a[2];
    b=a[1];
    a[4]=a[b];
    a[1]=a[0]+a[4];
    a[1]=a[1]+a[4];

    b=0;
    while (b<5)
    {
        cout << "a[" << b << "] = " << a[b] << endl;
        b=b+1;
    }
}
```

## Back to sorting

## Arrays and functions

- Functions and arrays work together in some odd ways.
  - Let's say we have an array "int B[8]"
  - First you pass the argument as "B".
  - You receive the parameter as "int X[]"

```
int bob(int X[])
{
    .....
}
.....
main
{
    int B[8];
    .....
    bob(B);
}
```

# More arrays and functions

- Further **unlike a normal variable** arrays passed as arguments **can** have their arguments change value.
  - This is because C/C++ make copies of normal variables when passing them.
  - But this is slow with large structures like arrays
  - So it skips the copy step and passes the array “by reference”
    - In other words it says “here is my array” passing a reference where to find it rather than passing a copy!

```
// Function calls with arrays
//

#include<iostream>
using namespace std;

void printList(int list[], int num)
{
    int i=0;
    while(i<num)
    {
        cout << "list[" << i << "] = " << list[i] << endl ;
        i=i+1;
    }
}
```

Page 1 of 3

```
void sort(int list[], int num)
{
    int i=0;
    int j=0;
    int tmp;
    while(i<num-1)
    {
        j=0;
        while(j<num-1)
        {
            if(list[j]>list[j+1])
            {
                tmp=list[j];
                list[j]=list[j+1];
                list[j+1]=tmp;
            }
            j=j+1;
        }
        i=i+1;
    }
}
```

Page 2 of 3

```
main()
{
    const int SIZE=8;
    int i;
    int A[SIZE]={1,4,-4,3,2,5,8,-11};

    printList(A,SIZE);
    cout << "----" << endl;
    sort(A,SIZE);
    printList(A,SIZE);
}
```

Page 3 of 3