Eng 101 – Day 7
1/21/04

Finish arrays, structures

Quiz on Monday

• Expect something similar to homework 1
  – Reading code and getting values
  – Will involve everything we’ve done so far, not including today.
  – May see something on the lab (how to compile, something like that)
• Quiz will be in the first 20 minutes of lecture.
  – Closed book, closed notes.

Today

• Finish arrays
• Introduce structures

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[]”

```c
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!

```c++
// 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;
    }
}

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;
    }
}

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);
}
```
It is sometimes useful to group different data structures into one unit.

So the "accounts receivable" at Michigan might have an int (say student number), and two doubles (balance, next payment)

```c++
struct account
{
    int sid;
    double balance;
    double next_payment;
};

main()
{
    account bob;
    bob.sid=100404;
    bob.balance=2013.23;
    bob.next_payment=bob.balance/12;
    cout << bob.next_payment<< endl;
}
```

```c++
#include<iostream>
using namespace std;

struct complex
{
    double real;
    double img;
};

complex c_add(complex a, complex b)
{
    complex result;
    result.real=a.real+b.real;
    result.img=a.img+b.img;
    return(result);
}

complex c_print(complex a)
{
    cout << "(" <<a.real << " + " << a.img << "i) ";
}

main()
{
    complex x ={1.0,3.0};
    complex y ={3.0, -1.0};
    complex z;
    z=c_add(x,y);
    c_print(x);
    cout << " + ";
    c_print(y);
    cout << " = ";
    c_print(z);
    cout << endl;
}
```
main()
{
    const int SIZE=4;
    complex x[SIZE];
    int i=0;

    while(i<SIZE)
    {
        x[i].real=i;
        x[i].img=i*(-i);
        i=i+1;
    }
    i=0;
    while(i<SIZE)
    {
        c_print(x[i]);
        cout << endl;
        i=i+1;
    }
}