

Arrays as parameters

- we've looked at basic use of arrays, and loops for going through the elements of an array
- it's common to want to pass an array to a function to let the array do the processing
- functions expecting array parameters designate this using [] and (usually) expecting the size as a second parameter, e.g.

```
void processArray(double arr[], int size)
```

- this allows us to use the same function for arrays of various sizes

Passing arrays to functions

- when passing an array we simply use it's name, e.g.

```
double myArray[20];  
fill(myArray, 20);
```

- unlike primitive data types (int, float, char, etc), the function can by default change the content of the array, so our fill function really can put data into our array
 - *(this is related to how arrays are internally represented and passed, we'll go into the why/how later in the course)*

Example: funcs to fill/print array

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

void fill(double arr[], int size);
void print(double arr[], int size);

int main()
{
    // create our array
    const int MaxSize = 50;
    double data[MaxSize];
    // pass to one function to fill it,
    // then the other function to print
    fill(data, MaxSize);
    print(data, MaxSize);
}
```

```
void fill(double arr[], int size)
{
    // read the (50) values from the user
    for (int p=0; p<size; p++) {
        cin >> arr[p];
    }
}

void print(double arr[], int size)
{
    // display the (50) values
    for (int p=0; p<size; p++) {
        cout << arr[p] << endl;
    }
}
```

Example: return sum of contents

```
// compute and return the sum of all the values in arr
float sum(float arr[], int size)
{
    float total = 0;
    for (int p=0; p<size; p++) {
        total = total + arr[p];
    }
    return total;
}
```

Example: copy an array

- copy content of source array into dest array
 - both arrays need to be at least as big as size passed
 - if dest array is too small this will overwrite memory past the end of the dest array (likely causing bugs when run)

```
void copy(long source[], long dest[], int size)
{
    for (int p=0; p<size; p++) {
        dest[p] = source[p];
    }
}
```