

---

## More Arrays

---

## Last Time

- ◆ We
  - Learnt how to pass arrays to functions
- ◆ Today we will
  - Start talking about searching arrays

---

## Arrays used as input

- ◆ What happens when we want to pass arrays to a function to be used only as output? We can't pass it by value...
  - `void large (int size, const int arry1[], const int arry2[], int arry3[]);`
- ◆ We can protect array arguments by putting `const` in front of them in prototype and function definition

---

## Problem

- ◆ Assume you have two arrays of doubles called `vals1` and `vals2`. They both contain `maxels` elements. Write a C++ `bool` function `identical` that will accept both arrays and return `true` if both arrays are identical; otherwise, return `false`. The call to your function might be by a statement of the following form:

```
if (identical (maxels, vals1, vals2))
    cout << "Arrays are identical" << endl;
else
    cout << "Arrays are not identical" << endl;
```

---

## Searching Arrays

- ◆ We search an array to find a particular element in an array.
- ◆ For example, we might like to search an array of student grades for all students who got higher than 90% (i.e. A's).
- ◆ How would we do this?

---

## Sequential or Linear Search

- ◆ Compare each element of the array with the value (or key) that we are searching for.
- ◆ This is called linear or sequential search.
- ◆ Linear Search Algorithm:
  - For each array element
    - If the current element contains the target
    - Return the subscript of the current element
  - Return -1

## Write the function findElement

```
int findElement(int [], int&);

int main()
{
    int grades[10];
    int element, index = -1;
    for(int i=0; i<10; i++)
        cin >> grades[i];
    cout << "Which element would you like to find?" << endl;
    cin >> element;
    findElement(grades, element, index);
    if(index == -1)
        cout << "Element could not be found!" << endl;
    else
        cout << "Element was found at index " << index <<
            endl;
}
```

11/12/04

CS150 Introduction to Computer Science 1

7

## Function to find element

```
void findElement(int ar[], int x,
                int & index)
{
    for(int i=0; i<10; i++)
        if(ar[i] == x)
            index = i;
}
```

11/12/04

CS150 Introduction to Computer Science 1

8

## Problem

- ◆ Write a function to return the index of the smallest element in a subarray.
- ◆ A subarray is a section of an array. The subarray is determined by its starting and ending indexes.
- ◆ The function will have the following arguments:
  - The array,
  - The starting index of the subarray,
  - The ending index of the subarray,
  - The index of the smallest element.

11/12/04

CS150 Introduction to Computer Science 1

9

## Function findIndexofMin

```
void findIndexofMin(const int x[], int
                   startIndex, int endIndex, int& index)
{
    index = startIndex;

    for(int i=startIndex + 1; i <= endIndex;
        i++)
        if(x[i] < x[index])
            index = i;
}
```

11/12/04

CS150 Introduction to Computer Science 1

10

## Summary

- ◆ In today's lecture we covered
  - Searching arrays
- ◆ Readings
  - Chapter 4

11/12/04

CS150 Introduction to Computer Science 1

11