Pointers

Lecture 3
Sections 9.1 - 9.3

Robb T. Koether

Hampden-Sydney College

Fri, Jan 19, 2018

Outline

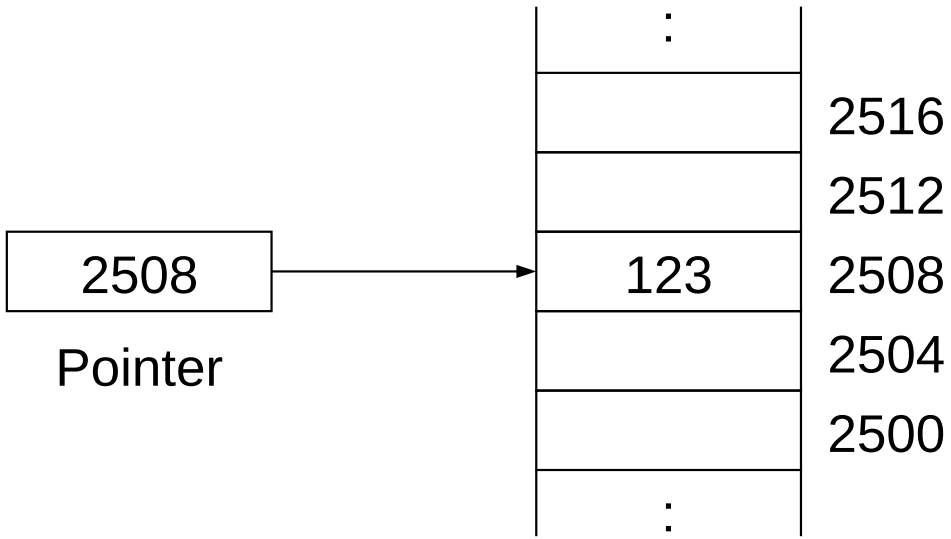
Pointers

Definition (Pointer)

A pointer is an object that holds the address of another object.

- On a 32-bit machine, all pointers occupy 4 bytes of memory.
- 4 bytes is sufficient to access 4GB of memory.
- On a 64-bit machine, all pointers occupy 8 bytes of memory.
- 8 bytes is sufficient to access 16,384 petabytes of memory.

Pointers



Declaring Pointers

Pointer Declaration

```
int* ptri;
float* ptrf;
```

- To declare a pointer, write the type of object that the pointer points to, followed by `*`, followed by the pointer name.

The Address Operator

Pointer Initialized to an Object

```
int i;
int* ptri = &i;  // &i is address of i
```

- The unary prefix address operator `&` returns the address of an object.
- The address may be assigned to a pointer.

Example

Example (Example)

- `Address Operator.cpp`

Null Pointers

Pointer Initialized to `NULL`

```
int* ptri = NULL;
```

- A null pointer is a pointer that has the value zero.
- The constant `NULL` has the value `0`.
- A null pointer does not point to any object.
- It is a good practice to initialize a pointer to `NULL` if it will not immediately point to an object.

Dereferencing Pointers

Dereferencing Pointers

```
int i = 10;
int* ptri = &i;                 // ptri points to i
cout << *ptri << endl;          // Print i
```

- The unary prefix dereference operator `*` is used to dereference a pointer.
- When a pointer is dereferenced, it returns the object that it points to.
- Never, ever dereference a null pointer.

Example

Example (Example)

- `Dereference Pointers.cpp`

Outline

Pointers as Function Parameters

- In CS I, we learn to pass objects to function *by reference*.
- The intention is to allow the object to be modified by the function.
- Before pass-by-reference was introduced, the same was accomplished by passing a pointer.
- To access the object form within the function, the pointer parameter must be dereferenced.

Example

Pointer Parameter

```
void swap(int* a, int* b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
    return;
}
```

Arrays as Function Parameters

- When an array is “passed” as a parameter, the name of the array is passed, as a pointer.
- Thus, the function receives a pointer to the first element of the array.
- This is far more efficient than copying the entire array.

Example

Array Parameter

```
void sort(int* list, int size)
{
    for (int i = 0; i < size - 1; i++)
    {
        int* q = list;
        while (q < list + size - 1)
        {
            if (*q > *(q + 1))
            swap(q, q + 1);
            q++;
        }
    }
    return;
}
```

Outline

Assignment

Homework

- Read Sections 9.1 - 9.3.