Pointers
Lecture 1
Section 2.4

Robb T. Koether
Hampden-Sydney College
Wed, Jan 14, 2009
Definition (Pointer)

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

- All pointers occupy 4 bytes of memory.
- 4 bytes is sufficient to access 4GB of memory.
Declaring Pointers

- **Pointer Declaration**

  ```
  int* pi;
  float* pf;
  ```

  Write the type of object that the pointer points to, followed by `*`, followed by the pointer name.
Null Pointers

A null pointer is a pointer that has the value zero.

A null pointer does not point to any object.

The constant `NULL` has the value `0`.

It is a good practice to initialize a pointer to `NULL` if it is not currently pointing to an object.
The Address Operator

The unary prefix address operator & returns the address of an object.

The address may be assigned to a pointer.

```
int i;
int* pi = &i;  // &i is address of i
```
Dereferencing Pointers

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

```c
int i = 10;
int* pi = &i; // pi points to i
cout << *pi << endl; // Print i
```
Demo - DereferencePointers.cpp

Run DereferencePointers.cpp
A pointer to `void` (`void* object`) is a pointer that does not point to any particular kind of object.

There is no such thing as a `void` "object."

`void` is treated as a neutral type for situations where the type does not matter or is unknown.

A `void` pointer may hold the address of any type of object.

A `void` pointer may not be dereferenced.
The pointer assignment $p1 = p2$ is legal if and only if:
- $p1$ and $p2$ point to the same type of object, or
- $p1$ is a pointer to `void`, or
- $p2$ is `NULL`.

In all other cases, we would have to cast $p2$ to be compatible with $p1$.

Assigning pointers to different types can cause serious problems.
What are the values of \( \star pi \) and \( \star pf \)?
Demo - PointerCompatibility.cpp

Run PtrCompatibility.cpp
Assignment

Homework

- Read Section 2.4, pages 63 - 71.
- Exercises 10, 11, 12, page 79.