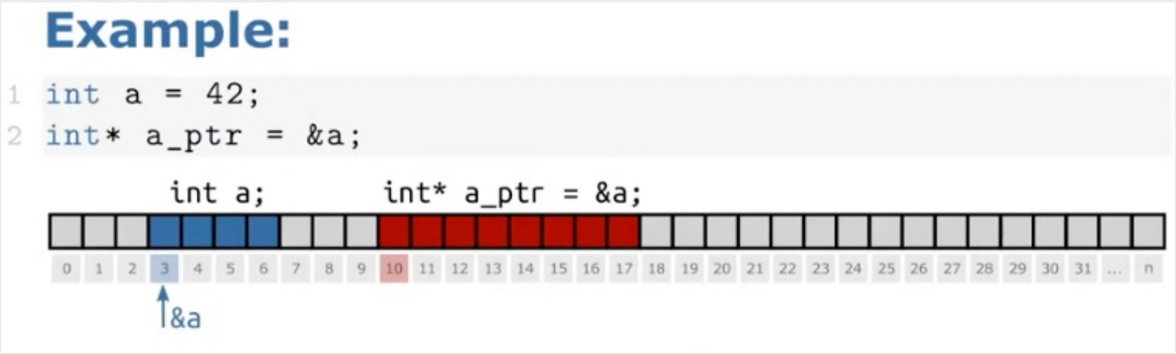
Raw Pointers

Address pointers for pointers

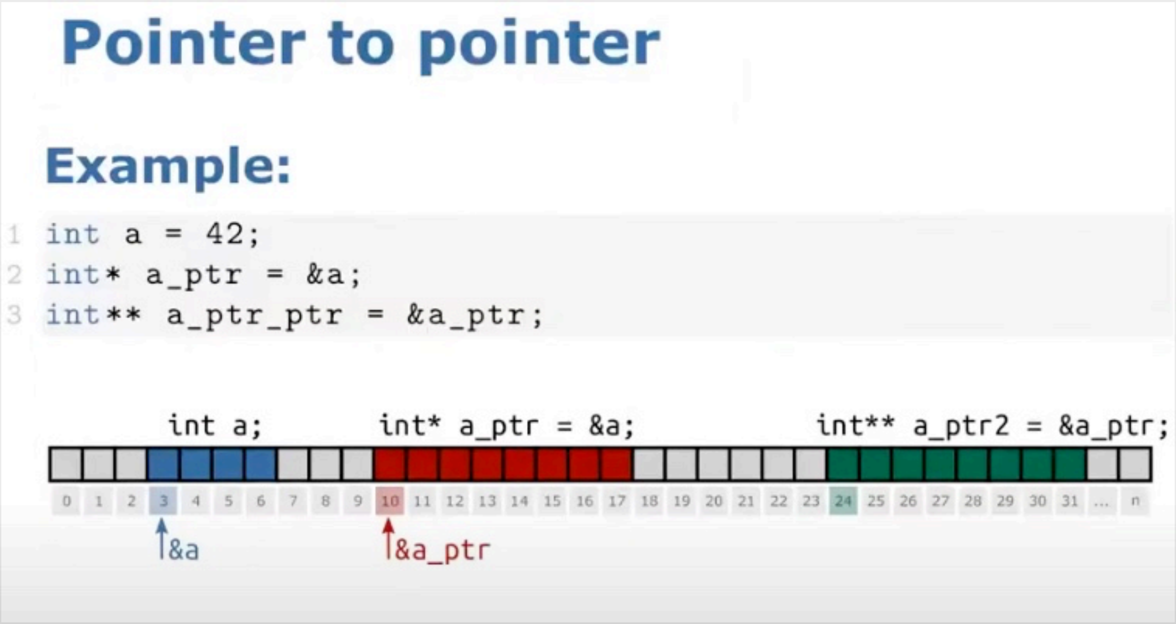
- Operator & returns the address of the variable in memory.
- Return value type is “pointer to value type”.
- sizeof(pointer) is 8 bytes in 64bit systems.



Example:

```
int a = 42;
int* a_ptr = &a;
```

int a;
int* a_ptr = &a;
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 ... n
&a



Pointer to pointer

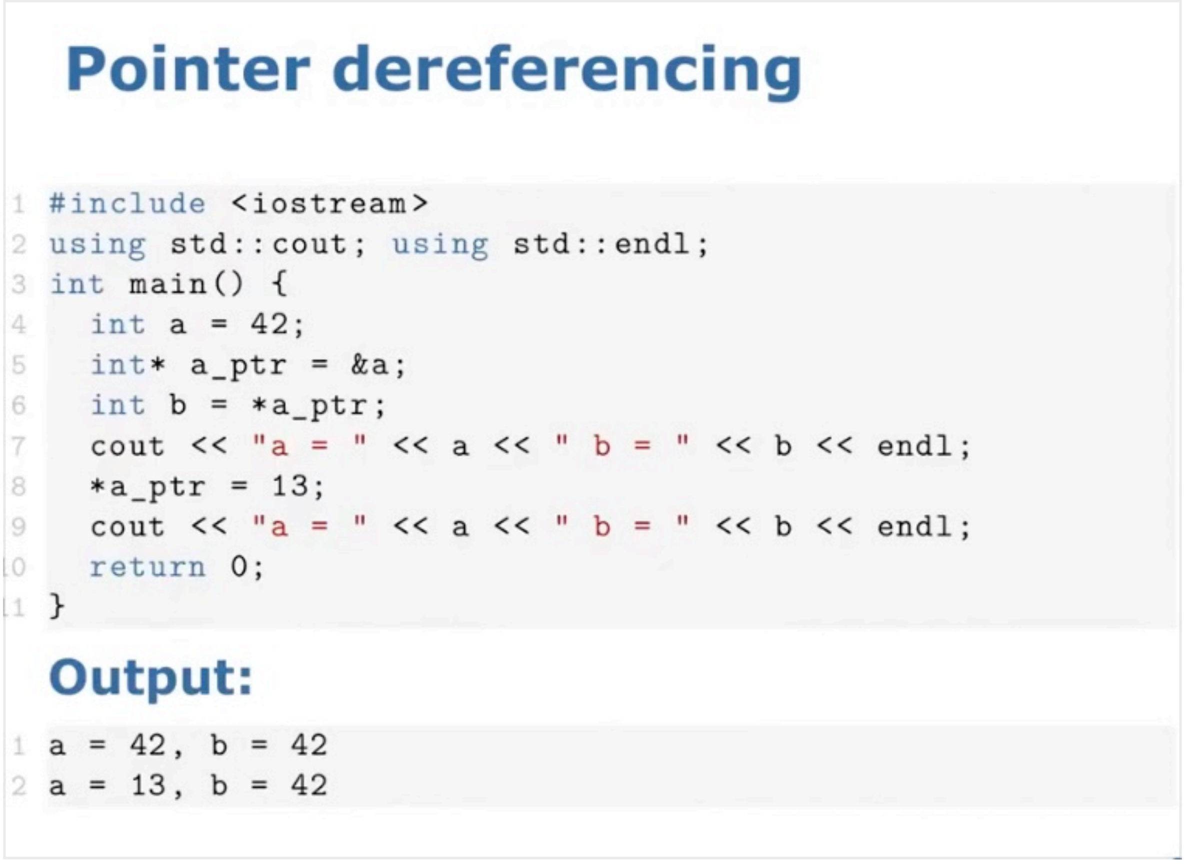
Example:

```
int a = 42;
int* a_ptr = &a;
int** a_ptr_ptr = &a_ptr;
```

int a;
int* a_ptr = &a;
int** a_ptr2 = &a_ptr;
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 ... n
&a
&a_ptr

Pointer Dereferencing

- Operator * returns the value of the variable to which the pointer points.
- Dereferencing of nullptr: Segmentation Fault
- Dereferencing of uninitialised pointer: Undefined behaviour

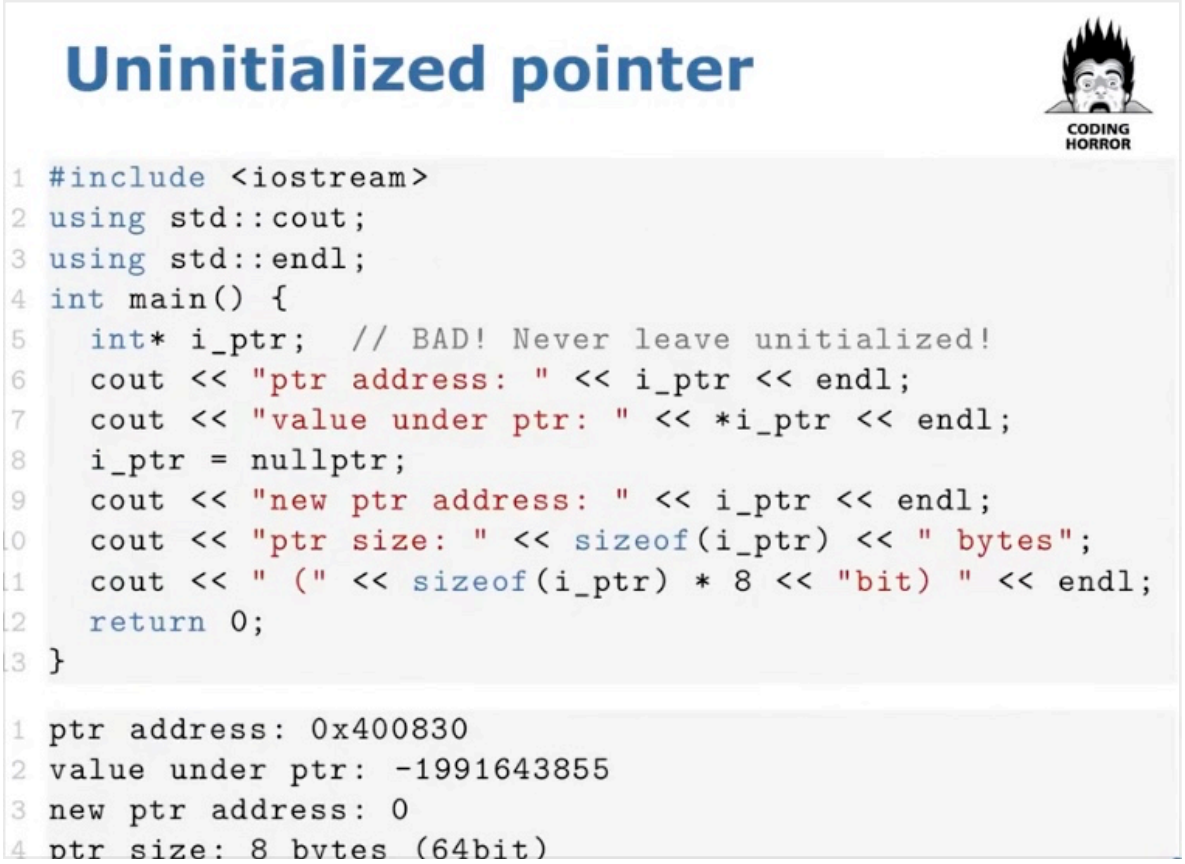


Pointer dereferencing

```cpp
#include <iostream>
using std::cout; using std::endl;
int main() {
  int a = 42;
  int* a_ptr = &a;
  int b = *a_ptr;
  cout << "a = " << a << " b = " << b << endl;
  *a_ptr = 13;
  cout << "a = " << a << " b = " << b << endl;
  return 0;
}
```

Output:

```
a = 42, b = 42
a = 13, b = 42
```



Uninitialized pointer

CODING HORROR

```cpp
#include <iostream>
using std::cout;
using std::endl;
int main() {
  int* i_ptr;  // BAD! Never leave unitialized!
  cout << "ptr address: " << i_ptr << endl;
  cout << "value under ptr: " << *i_ptr << endl;
  i_ptr = nullptr;
  cout << "new ptr address: " << i_ptr << endl;
  cout << "ptr size: " << sizeof(i_ptr) << " bytes";
  cout << " (" << sizeof(i_ptr) * 8 << "bit) " << endl;
  return 0;
}
```

```
ptr address: 0x400830
value under ptr: -1991643855
new ptr address: 0
ptr size: 8 bytes (64bit)
```