C++ Programming

Pointers 1

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

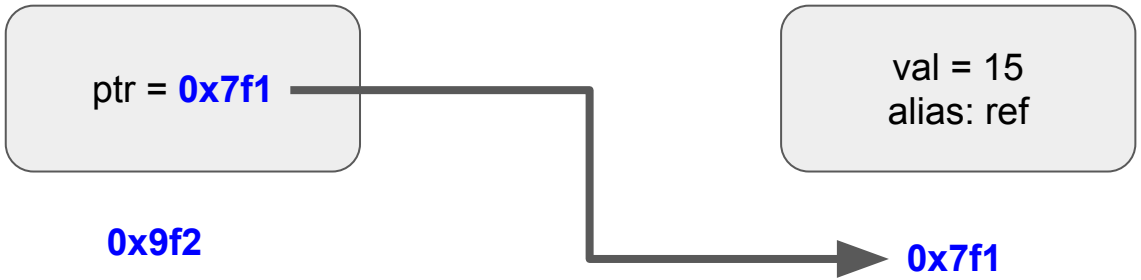
Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Pointers

```
int val = 15;
int &ref = val;

// 15 0x7f1
cout << val << " " << &val << "\n";
// 15 0x7f1
cout << ref << " " << &ref << "\n";

int *ptr = &val;
// 15 0x7f1 0x9f2
cout << *ptr << " " << ptr << " " << &ptr << "\n";
```



Pointers

- Pointer is a **variable** that store the address of another **variable**.
  - So there are **2 addresses** - one for **each variable**:
  - 1) The address it points toward it = content of the variable = **ptr**
  - 2) It is own address as a variable in memory = **&ptr**
  - How to get the value of the address I am pointing toward it? ***ptr**
- *ptr
  - Unary Operator
  - Name: Dereference Operator
  - Role: indirect addressing
  - We read it as:
    - value pointed to by

```
int val = 15;
int &ref = val;

// 15 0x7f1
cout << val << " " << &val << "\n";
// 15 0x7f1
cout << ref << " " << &ref << "\n";

int *ptr = &val;
// 15 0x7f1 0x9f2
cout << *ptr << " " << ptr << " " << &ptr << "\n";
```

Pointers

```
*ptr = 20;
// 20 20 20
cout << val << " " << ref << " " << *ptr << "\n";

int another = 30;
// 30 0x1afd
cout << another << " " << &another << "\n";

ptr = &another;
// 30 0x1afd 0x9f2       (0x9f2 did not change)
cout << *ptr << " " << ptr << " " << &ptr << "\n";

*ptr = 50;
// 20 50 50
cout << val << " " << another << " " << *ptr << "\n";

ptr = nullptr;  // point to nothing

if(!ptr)
    cout<<"NULL\n";
// NULL
```

- We can change 2 things:
  - The value where pointer is pointing into (*ptr)
    - Then the other variable is changed
  - The address of our pointer itself
- Whenever a pointer is not in use, assign nullptr
  - *ptr = RTE

Pointer initialization

```
double x = 10;
double* ptr1 = &x;

//double* ptr2 = NULL;  // C-style - OK

// C++11
double* ptr2 = nullptr; // better [preferred]
double* ptr3 = 0;       // ok also nullptr
//double* ptre = 1;     // CE: 0 or address

double *ptr4;   // Garbage - DON'T
```

- Initialization tip:
  - Either initialize to some **address**
  - Or set initially **nullptr**
  - Otherwise it is garbage!

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."