C++ Programming

Pointers and Functions

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)

Pass pointer by reference

```
27 int main() {
28
29     int x1 = 4;
30
31     do_math(x1);
32
33     cout << x1 << "\n"; // 20
34
35     x1 = 4;
36     do_math(&x1);
37
38     cout << x1 << "\n"; // 20
39
40     vector<int> v {5, 7, 8};
41     print(&v);
42
```

```
 5 void do_math(int &x) {
 6     // Guarantee this function
 7     // never called with null
 8     x = x + 1;
 9     x = x * 2;  // multiply with 2
10     x *= 2;     // multiply with 2
11 }
12
13 void do_math(int *x) {
14     if (x == nullptr)
15         return;
16
17     *x = *x + 1;
18     *x = *x * 2;    // multiply with 2
19     *x *= 2;        // multiply with 2
20 }
21
22 void print(vector<int> *ptr) {
23     for(auto v : *ptr)
24         cout<<v<<" ";
25 }
```

Return pointer

```
5  int* max(int *p1, int *p2) {
6      if (p1 == nullptr)      return p2;
7      if (p2 == nullptr)      return p1;
8      if (*p1 > *p2)
9          return p1;
10     return p2;
11 }
12
13 int* some() {
14     int temp = 10;
15     // NEVER. Local variable will be destroyed
16     return &temp;
17 }
18
19 int main() {
20     int x = 1, y = 5;
21     int *p = max(&x, &y);
22     cout << *p << "\n";
23
```

Pass pointer to reference

```
 3
 4  void hello(int &x) {
 5  }
 6
 7  int main() {
 8      int x = 1;
 9      int *p = &x;
10      int &y1 = x;
11
12      // int& needs a variable on the right side
13      int &y2 = *p;
14
15      hello(x);
16      hello(y2);
17      hello(*p);
18
```

Reference vs Pointer

```
 4 void fun1(int &x) {}
 5 void fun2(const int &x) {}
 6 void fun3(int *x) {}
 7
 8 int main() {
 9     int x = 10;
10     int *ptr = &x;
11
12     fun1(x);
13     //fun1(ptr);
14     fun1(*ptr);
15     //fun1(10);
16
17     fun2(x);
18     //fun2(ptr);
19     fun2(*ptr);
20     fun2(10);
21
22     //fun3(x);
23     fun3(&x);
24     fun3(ptr);
25     //fun3(10);
26
```

- Reference is *to some extent* safer than pointer
  - But still no guarantee
  - Common wrong statement: References cannot have a null value assigned
    - Doable with some workarounds
- fun2(10)
  - 10 creates temporary variable to be passed
  - Can only works in 2 cases
    - Void fun2(int x)
    - Void fun2(**const** int &x)

Tips

- In practice in many modern C++ software pointers are not frequent
  - STL plays a role in that (dynamic memory)
- Try to avoid using pointers if possible
  - E.g. is function(int &) is good enough?
    - If so, you don’t have to check the pointer = null
  - OOP Polymorphism depends on pointers/reference
- For small data types (e.g. integer), don’t try to use & to save memory
  - Not big deal
  - Void hello(int x) not void hello (const int &x);
- With heavy data (e.g. vector<person>), use & to avoid extra time/memory

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."