

# Range-based for Reference Random

CSE2013-17F

# Range-based for

- for-each-element

for ( for-range-declaration : expression )  
statement;

```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> vec;
    vec.push_back(100);
    vec.push_back(90);
    vec.push_back(95);

    for (int i = 0; i < vec.size(); i++)
        cout << vec[i] << endl;
}
```

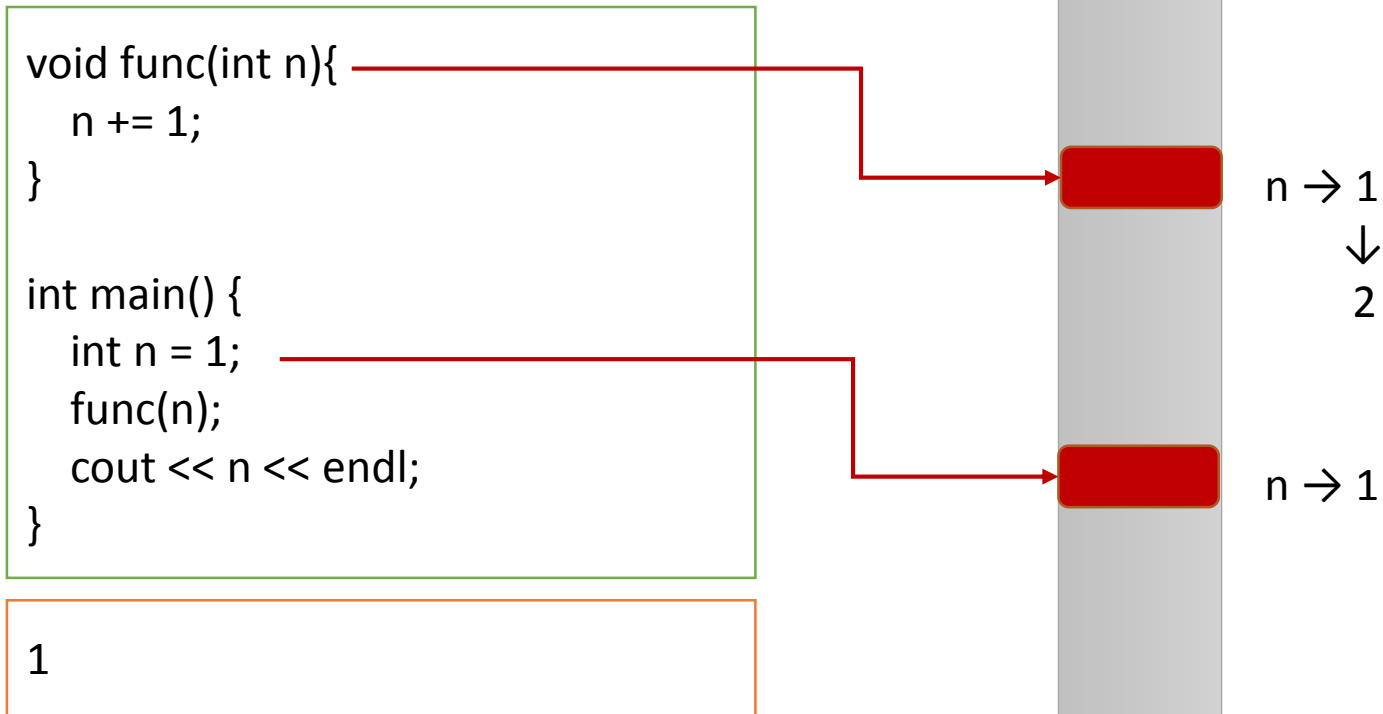
```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> vec;
    vec.push_back(100);
    vec.push_back(90);
    vec.push_back(95);

    for (auto n : vec)
        cout << n << endl;
}
```

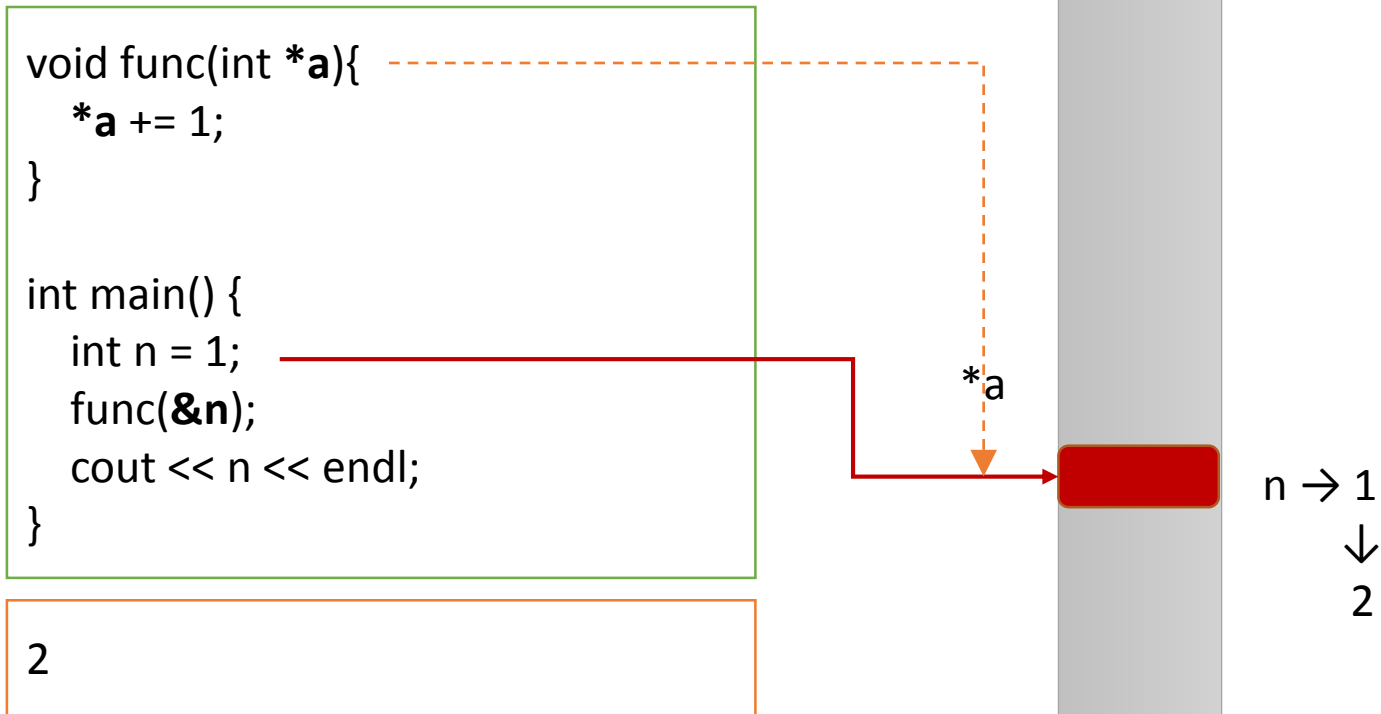
# Call-by-Value, Call-by-Reference

- Call-by-Value



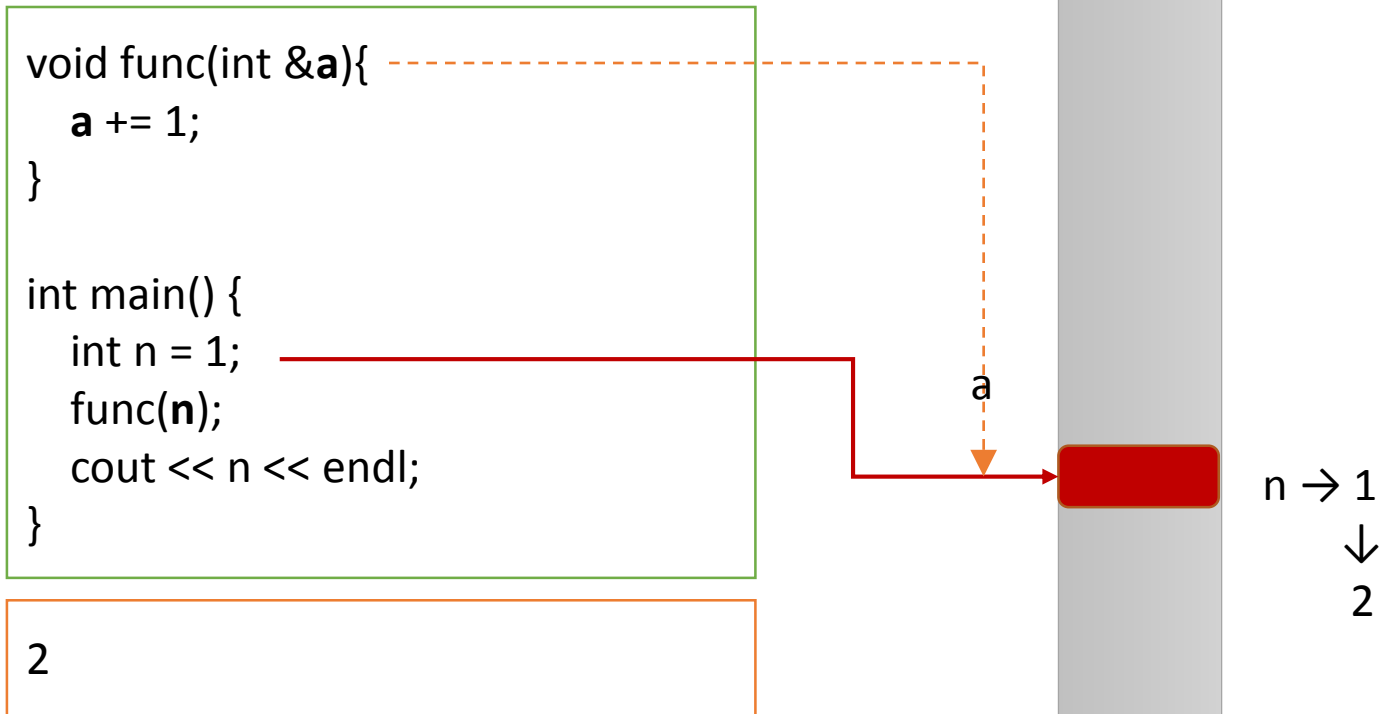
# Call-by-Value, Call-by-Reference

- Call-by-Reference ( C style - pointer)



# Call-by-Value, Call-by-Reference

- Call-by-Reference ( C++ style - reference)



# Reference

- Improving performance using reference
  - Use reference (pointer) instead of copying element

```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> vec;
    vec.push_back(100);
    vec.push_back(90);
    vec.push_back(95);

    for (auto n : vec)
        cout << n << endl;
}
```



```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> vec;
    vec.push_back(100);
    vec.push_back(90);
    vec.push_back(95);

    for (auto &n : vec)
        cout << n << endl;
}
```

# Swap

- C style (pointer)

```
void swap (int *a, int *b) {  
    int t = *a;  
    *a = *b;  
    *b = t;  
}  
  
int main() {  
    int x = 3, y = 5;  
    swap (&x, &y);  
    cout << x << '\t' << y << endl;  
}
```

- C++ style (reference)

```
void swap (int &a, int &b) {  
    auto t = a;  
    a = b;  
    b = t;  
}  
  
int main() {  
    int x = 3, y = 5;  
    swap (x, y);  
    cout << x << '\t' << y << endl;  
}
```

# Random Numbers

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main() {
    srand(3210); // set initial seed value
    for (int i = 0; i < 10; i++)
        cout << rand() << endl;
}
```

- Sequence of numbers from `rand()` is the same, if seed value is the same



# Random Numbers

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main() {
    srand(time(NULL));
    for (int i = 0; i < 10; i++)
        cout << rand() << endl;
}
```

- `time(NULL)` returns the number of seconds since 00:00, Jan 1, 1970 UTC