그런즉 너희가 먹든지 마시든지 무엇을 하든지 다 하나님의 영광을 위하여 하라 (고전 10:31)

NOTE: The following materials have been compiled and adapted from the numerous sources including my own. Please help me to keep this tutorial up-to-date by reporting any issues or questions. Send any comments or criticisms to idebtor@gmail.com Your assistances and comments will be appreciated.

Lab 3 - C++ 참조(Reference) 변수

단원

1. 참조 변수
2. 값으로 전달
3. 포인터로 전달
4. 참조로 전달
5. 참조로 반환

C++의 참조 변수는 C++ 프로그래밍 언어의 중요 개념입니다. 포인터만큼 강력하지는 않지만, 효율적인 프로그램을 작성할 수 있도록 도와준다.

1. 참조 변수

C++의 참조변수를 사용하면, 변수에 저장된 원래 데이터를 읽거나 수정하는데 사용할 수 있는 두번째 이름(또는 별칭)을 만들 수 있다. 즉, 다른 범위(Scope)내에 있더라도 참조를 선언하고 변수를 할당하면 두번째 이름을 사용하여(참조) **원래 변수의 값에 접근하고 수정할 수 있다**. 예를 들면, **함수 인수들을 참조변수로 선언하여** 변수를 참조하게 한 후 함수로 보내면 함수내에서도 효과적으로 원래 데이터를 수정 할 수 있다.

2. 값으로 전달

"전달"은 함수에게 인수를 제공하는 것이다. "값으로 전달"하는 과정이 가장 명확한 전달 방법이다. 함수가 호출이 되면, 함수내에 범위로 인수들이 새로 복사가 된다.

3. 포인터로 전달

"포인터로 전달"은 함수가 호출이 될 때 매칭되는 파라미터에 포인터를 인수로 전달하는 과정이다. 호출된 함수에서 **포인터 인수가 가리키는 변수의 값을 수정 할 수 있다**. 포인터로 전달하는 경우

포인터의 복사본이 함수로 전달된다. 그렇기에 함수에서 포인터를 수정할 경우, 복사된 포인터만 수정되기 때문에 원래 포인터는 수정되지 않고 기존 변수를 그대로 가리킨다.

4. 참조로 전달

"참조로"는 함수로 보내는 인수가 **이미 메모리에 존재하는 변수의 참조**이라는 뜻이다. 기존 변수의 독립된 복사본이 아니라는 뜻이다. 참조를 하는 것이기 때문에 참조변수에 가해지는 모든 연산은 그 변수가 참조하는 기존 변수를 직접 수정한다. 이것은 굉장히 **강력한** 개념인 이유는 함수에 큰 객체를 넘기는 것 보다 그 객체의(복사본이 아닌) 참조를 보내 **포인터로 보냈을 때 보다 간단한 과정을 거친다**. 참조변수는 절대 null 을 참조할 수 없기 때문에 모든 함수에 null 인지 확인 할 필요가 없다(포인터와 달리). 그리고, 힙(Heap)에 할당 된 것이 아니기에 메모리 관리에 오류가 날 걱정이 없다(할당 비할당 하는 과정이 없다).

5. 참조로 반환

함수에서 포인터를 반환하는 것처럼, **함수에서 참조를 반환할 수 있다**. 함수에서 참조를 반환하는 것은 암시된(implicit) 포인터를 반환한다. 그렇기에 함수에서 반환되는 참조는 **값을 할당하는 과정에서 좌편(할당 받는 변수) 위치**에 사용될 수 있다.

Step 1. 참조 변수

1. times.cpp 소스 파일을 생성하고, 아래 코드를 추가한다.

- list 에 있는 짝수 원소들에 10 을 곱한다.
- ranged-for-반복문과 참조 변수를 사용한다.
- 매크로 **#if 1:** #if 1 과 #else 사이에 코드를 사용하고, 나머지 코드는 무시한다.
- 매크로 **#if 0**: #else 와 #endif 사이에 코드를 사용하고, 위 코드를 무시한다.

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

#if 1
// mulitply even number elements in the list by 10
// without using reference variable.
int main(int argc, char *argv[]) {
   vector<int> list = { 0, 1, 2, 2, 4, 5, 6, 7, 8, 8, 10 };
   for (size_t i = 0; i < list.size(); i++) {

      cout << "your code here\n";

   }
   for (auto x: list) cout << x << " ";
   return 0;
}
```

```
#else

// mulitply even number elements in the list by 10
// using reference variable.
int main(int argc, char *argv[]) {
   vector<int> list = { 0, 1, 2, 2, 4, 5, 6, 7, 8, 8, 10 };

   cout << "your code here\n";

   for (auto x: list) cout << x << " ";
   return 0;
}
#endif
```

실행 예시:

```
$ g++ -std=c++11 times.cpp -o times
$ ./times
0 1 20 20 40 5 60 70 80 80 100
```

Step 2. 값으로 전달

1. setmax_val.cpp 소스 파일을 생성하고, 아래 코드를 추가한다.

- 값으로 전달한다.
- 주어진 배열을 이용해 가장 큰 값을 찾고, setmax() 함수에서 99 로 그 값을 수정한다.
- setmax()에서 수정되었음에도 main()에서 가장 큰 값이 수정되지 않는 점을 관찰한다. 그 이유는 의도적으로 함수가 변수를 값으로 전달하였기 때문이다.
- 추가적으로, 아래 코드처럼 모든 코드 예시에서 모르는 함수나 메소드(assert(), auto, etc.) 등, 스스로 찾아서 공부하길 바란다.

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

// gets the max value in the list and returns its index
int getmax(vector<int> vec) {
  assert(vec.size() > 0);
  auto max = vec[0];
  size_t idx = 0;
  for (size_t i = 0; i < vec.size(); i++) {
    cout << "your code here\n";
  }
  return idx;
}

// sets the max value in the list to 99
void setmax(vector<int> vec) {
  size_t idx = getmax(vec);
  cout << "your code here\n";
}

// With a given list, find the max value, then set it to 99 in setmax()
// In main(), we don't see the max value change that was made in setmax(),
// since functions are using pass-by-value on purpose.
int main(int argc, char *argv[]) {
  vector<int> list1 = {43, 10, 20, 75, 22, 33};
```

```
  vector<int> list2 = {33, 13, 45, 19, 39, 22};

  cout << ">list1: ";
  for (auto x: list1) cout << x << " ";
  cout << endl;
  setmax(list1);
  cout << "<list1: ";
  for (auto x: list1) cout << x << " ";
  cout << endl << endl;

  cout << ">list2: ";
  for (auto x: list2) cout << x << " ";
    cout << endl;
  setmax(list2);
  cout << "<list2: ";
  for (auto x: list2) cout << x << " ";
  cout << endl;
  return 0;
}
```

실행 예시:

```
$ g++ -std=c++11 setmax_val.cpp -o setmax_val
$ ./setmax_val
>list1: 43 10 20 75 22 33
<list1: 43 10 20 75 22 33

>list 2: 33 13 45 19 39 22
<list 2: 33 13 45 19 39 22
```

Step 3. 포인터로 전달

1. setmax_ptr.cpp 소스 파일을 생성하고, 아래 코드를 추가한다.

- 주어진 배열을 이용해 가장 큰 값을 찾고, setmax() 함수에서 99 로 그 값을 수정한다.
- setmax() 함수에 배열을 포인터로 전달한다.

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

// gets the max value in the list and returns its index
int getmax(vector<int> vec) {
   assert(vec.size() > 0);
   auto max = vec[0];
   size_t idx = 0;
   for (size_t i = 0; i < vec.size(); i++) {
      cout << "your code here\n";
   }
   return idx;
}

// sets the max value in the list to 99
// your code here - define setmax() here
```

```
// With a given list, find the max value, then set it to 99 in setmax()
int main(int argc, char *argv[]) {
   vector<int> list1 = {43, 10, 20, 75, 22, 33};
   vector<int> list2 = {33, 13, 45, 19, 39, 22};

   cout << ">list1: ";
   for (auto x: list1) cout << x << " "; cout << endl;
   cout << “your code here – invoke setmax()” << endl;
   cout << “<list1: “;
   for (auto x: list1) cout << x << “ “;
   cout << endl << endl;

   cout << ">list2: ";
   for (auto x: list2) cout << x << " "; cout << endl;
   cout << “your code here – invoke setmax()” << endl;
   cout << “<list2: “;
   for (auto x: list2) cout << x << “ “;
   cout << endl;

   return 0;
}
```

실행 예시:

```
$ g++ -std=c++11 setmax_ptr.cpp -o setmax_ptr
$ ./setmax_ptr
>list1: 43 10 20 75 22 33
<list1: 43 10 20 99 22 33

>list 2: 33 13 45 19 39 22
<list 2: 33 13 99 19 39 22
```

Step 4. 참조로 전달

1. setmax_ref.cpp 소스 파일을 생성하고, 아래 코드를 추가한다.

- 주어진 배열을 이용해 가장 큰 값을 찾고, setmax() 함수에서 99 로 그 값을 수정한다.
- setmax() 함수를 코딩하는 것으로 이번 단계는 끝난다.
- setmax() 함수에 있는 배열은 *참조로 전달* 되었다.

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

// gets the max value in the list and returns its index
int getmax(vector<int> vec) {
   assert(vec.size() > 0);
   auto max = vec[0];
   size_t idx = 0;
   for (size_t i = 0; i < vec.size(); i++) {
      cout << "your code here\n";
   }
   return idx;
```

```
}

// sets the max value in the list to 99
// your code here - define setmax() here

// With a given list, find the max value, then set it to 99 in setmax()
int main(int argc, char *argv[]) {
   vector<int> list1 = {43, 10, 20, 75, 22, 33};
   vector<int> list2 = {33, 13, 45, 19, 39, 22};

   cout << ">list1: ";
   for (auto x: list1) cout << x << " ";
   cout << endl;
   setmax(list1);
   cout << "<list1: ";
   for (auto x: list1) cout << x << " ";
   cout << endl << endl;

   cout << ">list2: ";
   for (auto x: list2) cout << x << " ";
   cout << endl;
   setmax(list2);
   cout << "<list2: ";
   for (auto x: list2) cout << x << " ";
   cout << endl;

   return 0;
}
```

실행 예시:

```
$ g++ -std=c++11 setmax_ref.cpp -o setmax_ref
$ ./setmax_ref
>list1: 43 10 20 75 22 33
<list1: 43 10 20 99 22 33

>list 2: 33 13 45 19 39 22
<list 2: 33 13 99 19 39 22
```

Step 5. 참조로 반환

1. setmax_ret.cpp 소스 파일을 생성하고, 아래 코드를 추가한다.

- 주어진 배열을 이용해 가장 큰 값을 찾고, 99 로 그 값을 수정한다.
- getmax()를 코딩하여 가장 큰 값을 가진 원소를 **참조로 반환** 한다. 단, 원소의 인덱스(index)는 반환하지 않는다.
- setmax()를 코딩하여 가장 큰 값을 가진 원소가 참조가 반환된 것을 사용 하도록 한다. **함수**에서 반환된 참조는 **값을 할당하는 과정에서 좌편(할당 받는 변수) 위치**에 사용될 수 있다는 점을 기억한다.
- list 배열이 getmax()와 setmax()의 참조로 전달되도록 코딩한다.

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

// gets the max value in the list and returns its index
// your code here - define getmax() here

// sets the max value in the list to 99
// your code here - define setmax() here

// With a given list, find the max value, then set it to 99 in setmax()
int main(int argc, char *argv[]) {
   vector<int> list1 = {43, 10, 20, 75, 22, 33};
   vector<int> list2 = {33, 13, 45, 19, 39, 22};

   cout << ">list1: ";
   for (auto x: list1) cout << x << " ";
   cout << endl;
   setmax(list1);
   cout << "<list1: ";
   for (auto x: list1) cout << x << " ";
   cout << endl << endl;

   cout << ">list2: ";
   for (auto x: list2) cout << x << " ";
   cout << endl;
   setmax(list2);
   cout << "<list2: ";
   for (auto x: list2) cout << x << " ";
   cout << endl;

   return 0;
}
```

실행 예시:

```
$ g++ -std=c++11 setmax_ret.cpp -o setmax_ret
$ ./setmax_ret
>list1: 43 10 20 75 22 33
<list1: 43 10 20 99 22 33

>list 2: 33 13 45 19 39 22
<list 2: 33 13 99 19 39 22
```

Step 6. touppers.cpp

C/C++ 언어에서 제공하는 toupper() 함수에서는 문자를 대문자로 변환한다. 하지만, 단어나 문자열은 안된다. touppers()라는 함수를 생성하여 추가적인 메모리를 사용하지 않고, 문자열을 모두 대문자로 변환한다.

```
Use a skeleton file, 'touppers.cpp', provided.
Use a reference variable and 'toupper()', but not functions in <algorithm>.
```

실행 예시:

```
$ g++ -std=c++11 touppers.cpp -o touppers
```

```
$ ./touppers
Enter words: Hello Mr. Kim
HELLO MR. KIM
```

제출파일 목록

- times.cpp
- setmax_val.cpp
- setmax_ptr.cpp
- setmax_ref.cpp
- setmax_ret.cpp
- touppers.cpp

One thing I know, I was blind but now I see. John 9:25