### **C++ 레퍼런스(참조자)에 대한 개념 정리**

#### **기본 개념**

* **레퍼런스**: 변수의 별명으로, 특정 변수에 대한 또 다른 이름을 제공.

cpp

* int& another\_a = a; // 변수 a의 또 다른 이름 another\_a 선언

#### **레퍼런스 선언 규칙**

* 레퍼런스를 정의할 때는 반드시 초기화(누구의 별명인지 명시)가 필요.

cpp

* int& another\_a = a; // 초기화 필요  
  int& another\_a; // 불가능
* **한번 별명이 되면 다른 변수의 별명으로 변경 불가**.

#### **메모리 할당 및 불필요한 연산자 생략**

* 레퍼런스는 항상 메모리가 할당되지는 않음.
* 불필요한 &와 \*를 사용할 필요 없음.

#### **상수에 대한 참조**

* 상수에 대한 참조자는 가능.

cpp

* const int &ref = 4; // 상수 참조 가능  
  a = ref; // a = 4와 동일
* int &ref = 4; 같은 것은 불가능 (상수를 비상수 레퍼런스로 참조 불가).

#### **리터럴**

* 소스코드 상에서 고정된 값을 가지는 것.
* 문자열 리터럴: 큰 따옴표로 묶인 문자열. 예: "good"
* 리터럴은 프로그램 실행 시 보관될 공간이 생성되며 변경 불가.

#### **레퍼런스와 배열**

* **레퍼런스의 배열은 불가능**.

cpp

* int& arr[2] = {a, b}; // 불가능
* **배열의 레퍼런스는 가능**.

cpp

* int (&ref)[3][2]; // [3][2]의 2차원 int 배열의 레퍼런스

#### **레퍼런스를 리턴하는 함수**

* 함수에서 지역 변수의 레퍼런스를 리턴하면 위험 (dangling 레퍼런스).

cpp

* int& func() {  
   int a = 2;  
   return a; // 위험: a는 함수 종료와 함께 소멸  
  }

#### 외부 참조자를 리턴하는 경우

* 함수가 외부 참조자를 리턴하면 효율적으로 주소값 복사로 전달 가능.

cpp

* int& function(int& a) {  
   a = 5;  
   return a;  
  }

#### 리턴 값을 참조자로 받기

* 리턴하는 값이 레퍼런스가 아닐 경우, 문장 종료 후 소멸되어 댕글링 레퍼런스 발생.
* 예외적으로 **const 레퍼런스로 받으면 사용 가능**.