

Day-32

Constants and Literals, const

Constants and Literals

- Constant(상수)는 컴파일 타임에 평가할 수 있는 표현식
- literal(리터럴)은 상수에 할당할 수 있는 고정된 값
 - 리터럴은 소스 코드에 포함된 상수 값을 나타내는 C++ 토큰
- Literal types(리터럴 타입)
 - 스칼라 타입의 구체적 값 `bool`, `char`, `int`, `float`, `double`
 - `const char[]` 타입의 문자열 리터럴
 - `nullptr`
 - 사용자-정의 리터럴

const Keyword

- `const` 키워드는 초기화 후 값을 변경하지 않는 객체를 나타냄
- 선언할 때 초기화 되어야 함

```
const int SIZE = 3;
// SIZE = 4; // 컴파일 에러(SIZE는 const)

int size = 3;
const int size2 = size;
int C[size2] = {1, 2, 3}; // 좋지 않은 프로그래밍
// 몇몇 컴파일러는 가변 크기 스택 배열을 허용 -> 위험
```

const Keyword and Pointer

- `int*` → `const int*`
- `const int*` → (불가능) `int*`

```
void f1(const int* arr){} // 배열의 값은 수정 불가

void f2(int* arr){}

int * ptr = new int[3];
const int* cptr = new int[3];
f1(ptr);      // ok
f2(ptr);      // ok
f1(cptr);     // ok
// f2(cptr); // 컴파일 에러
```

- `int*` 포인터를 `int`로
 - 포인터의 값을 수정할 수 있음
 - 포인터가 참조하는 요소를 수정할 수 있음
- `const int*` 포인터는 `const int`에 대한 포인터 : `(const int)*`로 읽기
 - 포인터의 값을 수정할 수 있음
 - 포인터가 참조하는 요소는 수정할 수 없음
- `int *const` `const` 포인터를 `int`로
 - 포인터의 값은 수정할 수 없음
 - 포인터가 참조하는 요소는 수정할 수 있음
- `const int *const` 포인터를 `const int`로
 - 포인터의 값을 수정할 수 없음
 - 포인터가 참조하는 요소는 수정할 수 없음
- 참고 : `const int*`(서양식 표기법)은 `int const*`(동양식 표기법)와 동일