

컴퓨터공학 All in One

C/C++ 문법, 자료구조 및 심화 프로젝트 (나동빈)
제 45강 - C언어와 C++ 비교하기

학습 목표

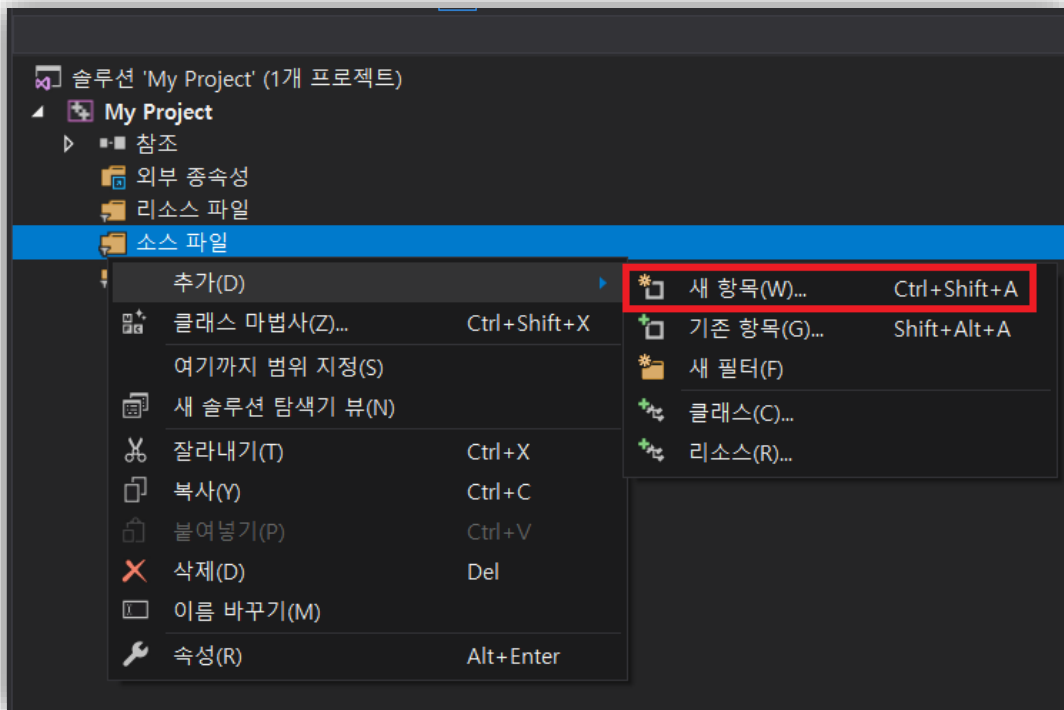
C언어와 C++ 비교하기

- 1) C언어와 C++ 차이점을 이해하고 C++로 'Hello World'를 출력할 수 있습니다.
- 2) C++에서의 문자열 자료형 및 메모리 동적 할당의 개념에 대해서 이해할 수 있습니다.

C언어와 C++ 비교하기

C++의 Hello World

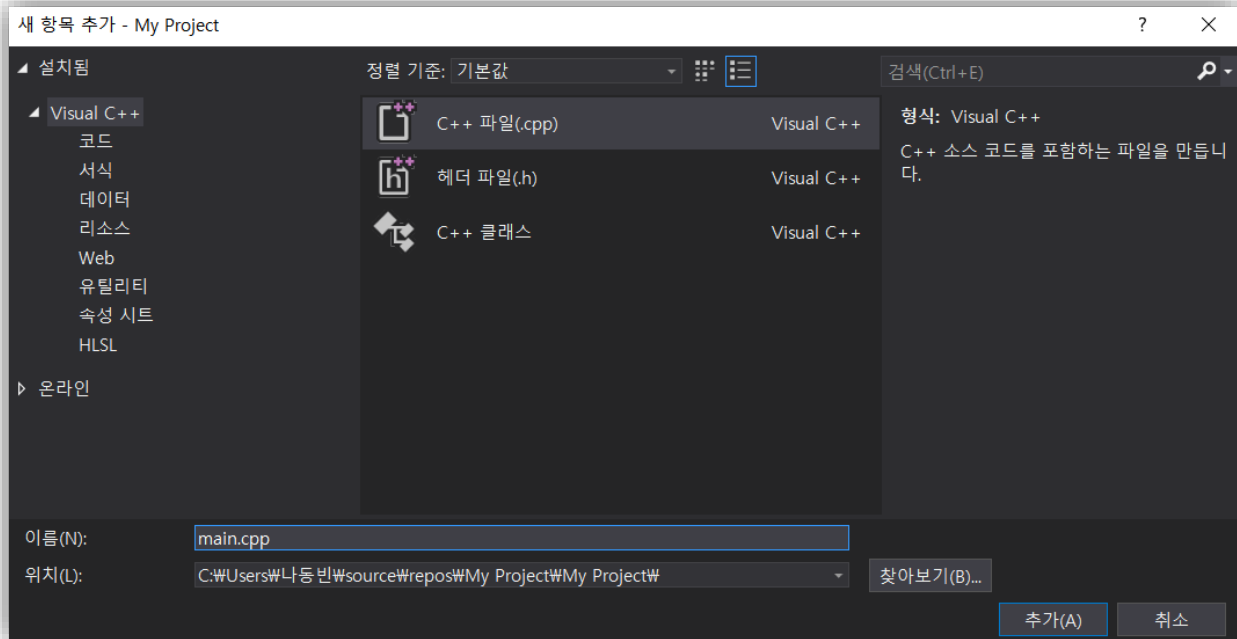
‘소스 파일’ - ‘추가’ - ‘새 항목’



C언어와 C++ 비교하기

C++의 Hello World

특정한 이름으로 .cpp 파일 생성



C언어와 C++ 비교하기

C++의 Hello World

```
#include <iostream>

using namespace std;

int main(void) {
    cout << "Hello World" << endl;
    system("pause");
    return 0;
}
```

C언어와 C++ 비교하기

C++의 iostream

iostream 라이브러리는 C++ 표준 입출력 라이브러리입니다. C언어의 `stdio.h`와 흡사하게 사용됩니다. 과거에는 `iostream.h`로 사용이 되었지만 최신 C++ 문법에서는 `.h`를 붙이지 않습니다.

C언어와 C++ 비교하기

C++의 기본 입출력

C언어에서는 `printf()`, `scanf()` 함수에서 형식 지정자를 적어주어야 했으나 C++에서는 형식 지정자를 넣어주지 않아도 변수를 타입에 맞게 적절히 입출력을 해줍니다.

C언어와 C++ 비교하기

C++의 기본 입출력

```
#include <iostream>
#include <string>

int main(void) {
    std::string input;
    std::cin >> input;
    std::cout << input << std::endl;
    system("pause");
    return 0;
}
```


C언어와 C++ 비교하기

C++의 기본 입출력

C++ 기본 입출력 라이브러리에서는 연산자 `>>`와 `<<`를 제공합니다. 이를 활용하여 모든 기본 자료형을 입출력 할 수 있습니다. 특히 입력을 받는 연산자 `>>`는 공백 문자(Space, Enter, Tab)을 기준으로 입력을 받습니다.

C언어와 C++ 비교하기

C++의 네임스페이스

네임스페이스(Namespace)는 특정한 영역에 이름을 설정할 수 있도록 하는 문법입니다. 네임스페이스는 서로 다른 개발자가 공동으로 프로젝트를 진행할 때 각자 개발한 모듈을 하나로 합칠 수 있도록 해줍니다.

C언어와 C++ 비교하기

C++의 네임스페이스

```
#include <iostream>

namespace A {
    void function() {
        std::cout << "A Namespace" << std::endl;
    }
}

namespace B {
    void function() {
        std::cout << "B Namespace" << std::endl;
    }
}

int main(void) {
    A::function(); // 범위 지정 연산자(::)
    B::function();
    system("pause");
    return 0;
}
```

C언어와 C++ 비교하기

C++의 네임스페이스

using 키워드를 이용하여 표준 라이브러리(std)를 모두 사용하도록 처리할 수 있습니다.

```
#include <iostream>
#include <string>

using namespace std;

int main(void) {
    string input;
    cin >> input;
    cout << input << endl;
    system("pause");
    return 0;
}
```

C언어와 C++ 비교하기

C++의 문자열 자료형

C++은 표준 문자열 자료형을 제공합니다. string 헤더 파일에 정의되어 있습니다.

C언어의 문자열: `char arr[SIZE];`

C++의 문자열: `string s;`

C언어와 C++ 비교하기

C++의 문자열 자료형

```
#include <iostream>
#include <string>

using namespace std;

int main(void) {
    string input;
    cin >> input;
    for (int i = 0; i < input.size(); i++) {
        cout << input[i] << '\n';
    }
    system("pause");
    return 0;
}
```

C언어와 C++ 비교하기

C++의 문자열 자료형

C++에서 공백을 포함하여 한 줄을 모두 문자열 형태로 입력 받고자 한다면 `getline()` 함수를 사용할 수 있습니다.

```
#include <iostream>
#include <string>

using namespace std;

int main(void) {
    string input;
    getline(cin, input);
    for (int i = 0; i < input.size(); i++) {
        cout << input[i] << '\n';
    }
    system("pause");
    return 0;
}
```

C언어와 C++ 비교하기

C++의 문자열 자료형

C++의 string은 다른 자료형으로의 변환이 간편합니다.

```
#include <iostream>
#include <string>

using namespace std;

int main(void) {
    int i = 123;
    string s = to_string(i);
    cout << "정수 -> 문자열: " << s << endl;
    s = "456";
    i = stoi(s);
    cout << "문자열 -> 정수: " << i << endl;
    system("pause");
    return 0;
}
```


C언어와 C++ 비교하기

C++의 동적 할당

```
#include <iostream>
#define SIZE 100

using namespace std;

int *arr;

int main(void) {
    arr = new int[SIZE]; // 동적 할당
    for (int i = 0; i < SIZE; i++) {
        arr[i] = i;
    }
    for (int i = 0; i < SIZE; i++) {
        cout << arr[i] << ' ';
    }
    delete arr; // 할당 해제
    system("pause");
    return 0;
}
```

C언어와 C++ 비교하기

C언어와 C++ 비교하기

- 1) C++는 객체 지향 패러다임을 따르고 있는 언어이지만 C언어는 절차적 프로그래밍 언어입니다.
- 2) 다시 말해 C++은 객체 중심의 언어이며 C언어는 함수 기반의 언어입니다.
- 3) C++은 C언어의 구조체(Struct) 대신에 클래스(Class)를 사용합니다.
- 4) C++는 공식적으로 예외 처리(Exception Handling) 기술을 지원합니다.

배운 내용 정리하기

C언어와 C++ 비교하기

- 1) C언어와 C++는 문법적으로 매우 흡사한 구조를 가지고 있습니다.
- 2) C++에서는 다양한 라이브러리를 제공하므로 더 짧고 효율적인 코딩이 가능합니다.
- 3) 가장 많이 사용되는 표준 라이브러리 중 하나는 string입니다.