

프로그래밍 역량 강화 전문기관, 민코딩

Visual Studio 활용



학습을 위한 최소 Cpp 기본 문법 리뷰

필수 기본 문법 1

✓cin, cout

✓if, for, 변수, 배열, 함수

필수 기본 문법 2

- ✓동적배열, 정적배열
- ✓reference
- ✓foreach 문

Refactoring Tool 세팅

실습에 필요한 추가 기능

✓ReSharper C++

- C++ 개발자를 위한 Visual Studio 유료 Extension (교육시에만 무료 라이선스)
- <https://www.jetbrains.com/ko-kr/resharper-cpp/>

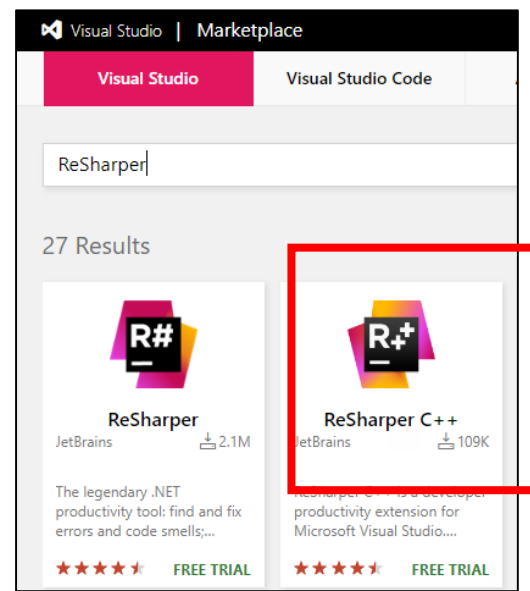
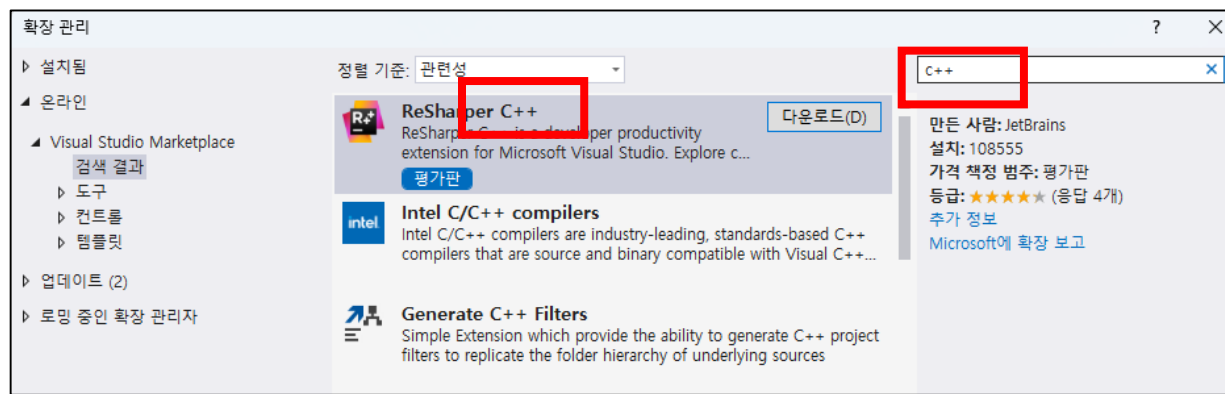
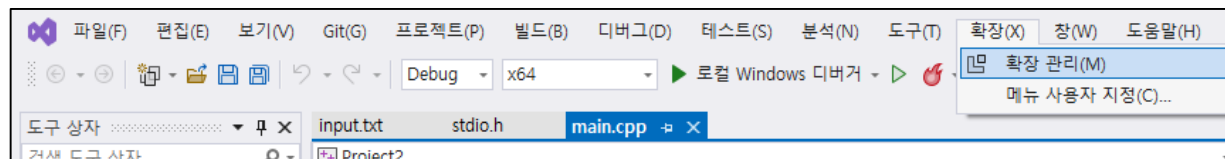
✓주요 기능

- 소스코드의 문제점에 대한 오류를 알려주고, 자동 해결 기능이 추가 됨
 - Visual Studio 기존 기능보다 더 세심한 기능
- 코드 리팩터링 기능
 - Visual Studio 기존 기능보다 더 많은 기능 제공

Extention 추가

✓ Visual Studio Market Place

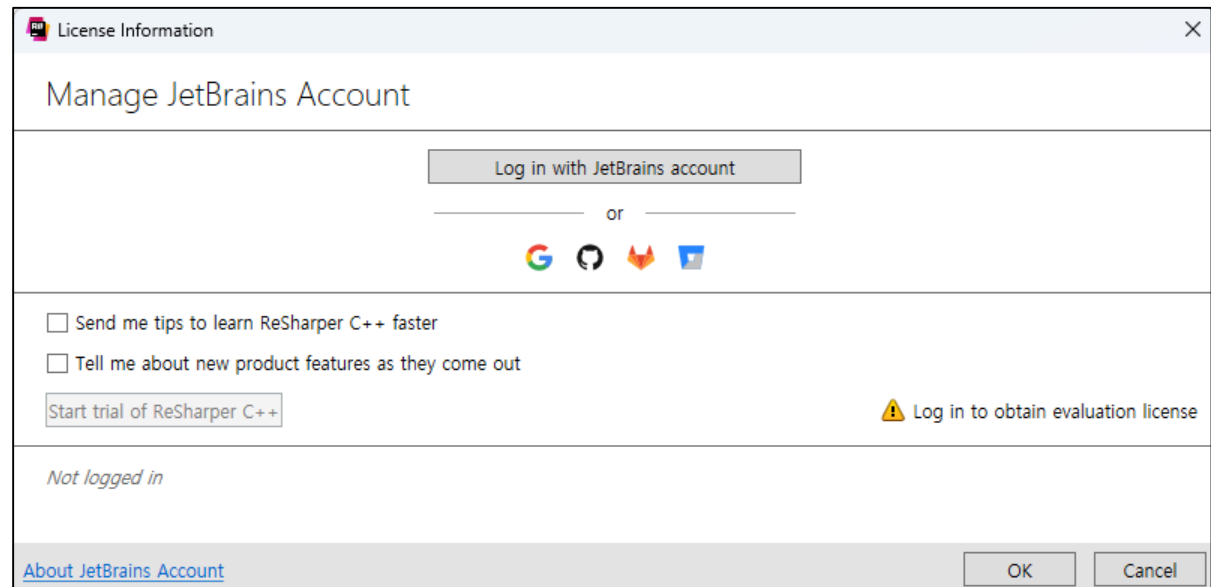
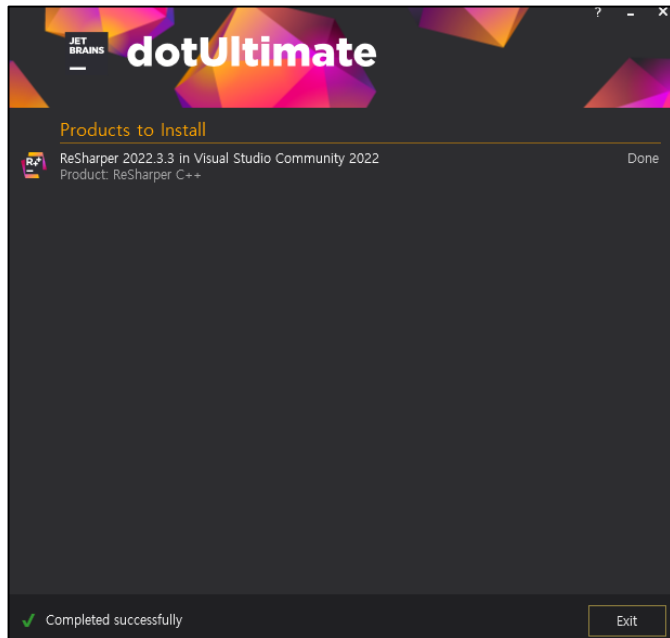
- 사외 : 확장 (Extention) > 확장 관리 에서 C++로 검색 후 평가판 설치
- 사내 : <https://marketplace.visualstudio.com/> 에서 ReSharper C++ 검색 후 다운&설치



ReSharper (C#버전) 이 아닌, ReSharper C++ 버전

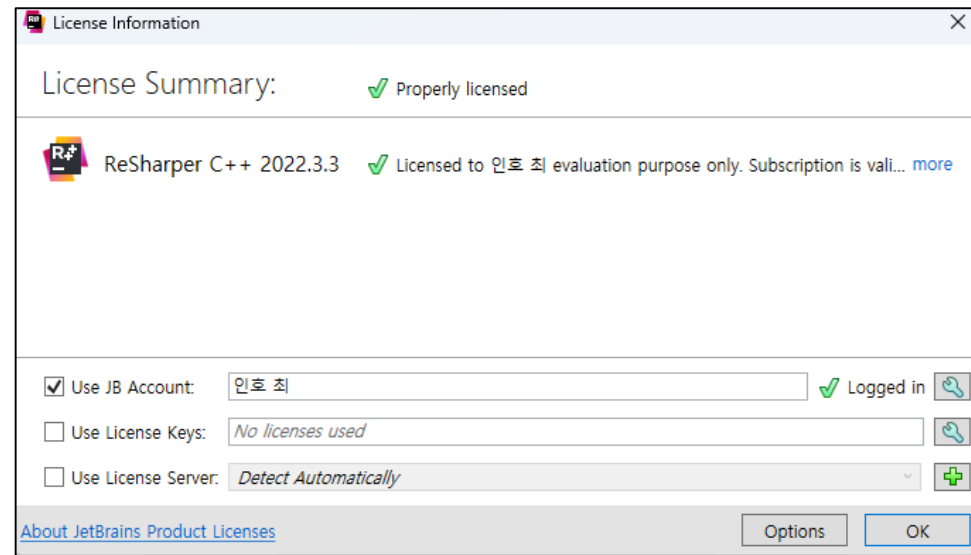
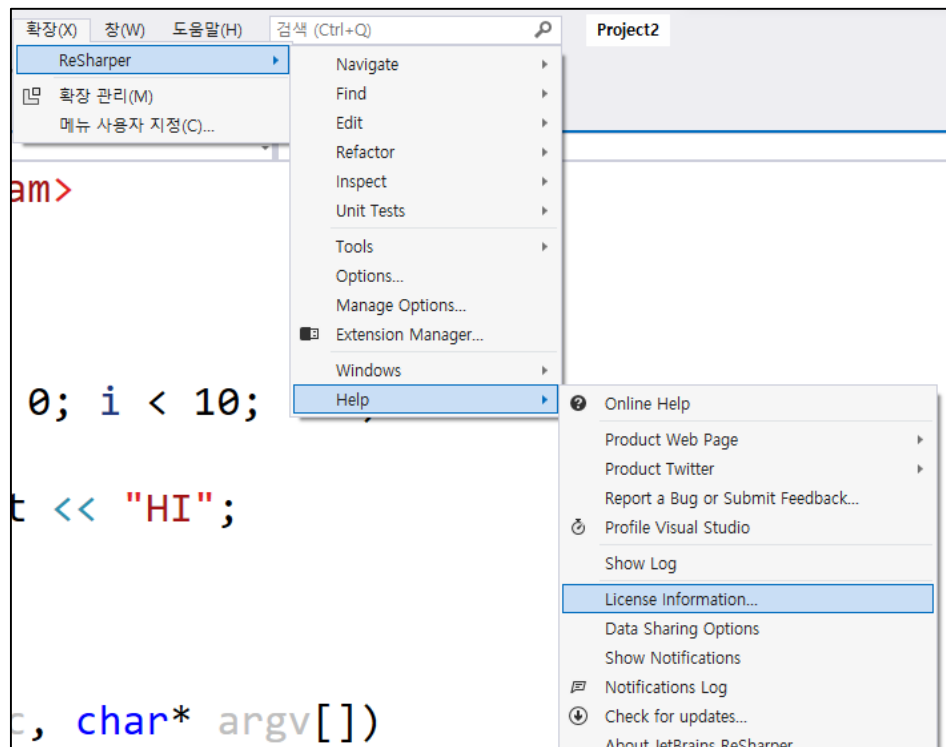
설치하기

- ✓ Visual Studio를 끄고, 설치 완료 후 다시 실행하기
 - 가입 7 로그인 하면 Refactoring Tool 사용 시작 가능



설치 확인

- ✓확장 > ReSharper 메뉴가 있어야 한다.
 - + 라이선스 확인



[참고] ReSharper C++ 라이선스

✓영구 풀백 라이선스

- 구독형 이지만, 12개월 구독이 끝나면 마지막 버전에 대해 영구 라이선스
- 가격 : \$350 (약 45만원)

✓30일 이하 교육에는 Free License 사용 가능

Free License Programs[Academic Licensing](#)[Open Source](#)[User Groups](#)[Events Partnership](#)[Developer Recognition](#)

FAQ

What are the licensing options for course instructors?

Our courses last less than 1 month. Can we participate in the Training Support Program?

No, if your courses take 30 days or less to complete, we suggest that you use the free 30-day trial versions of our software.

How can students renew their subscriptions once the training is over?

What if we run out of coupon codes during the year?

What happens if very few coupons are activated by students?

How can a student activate a coupon code?

How long is a coupon code valid for?

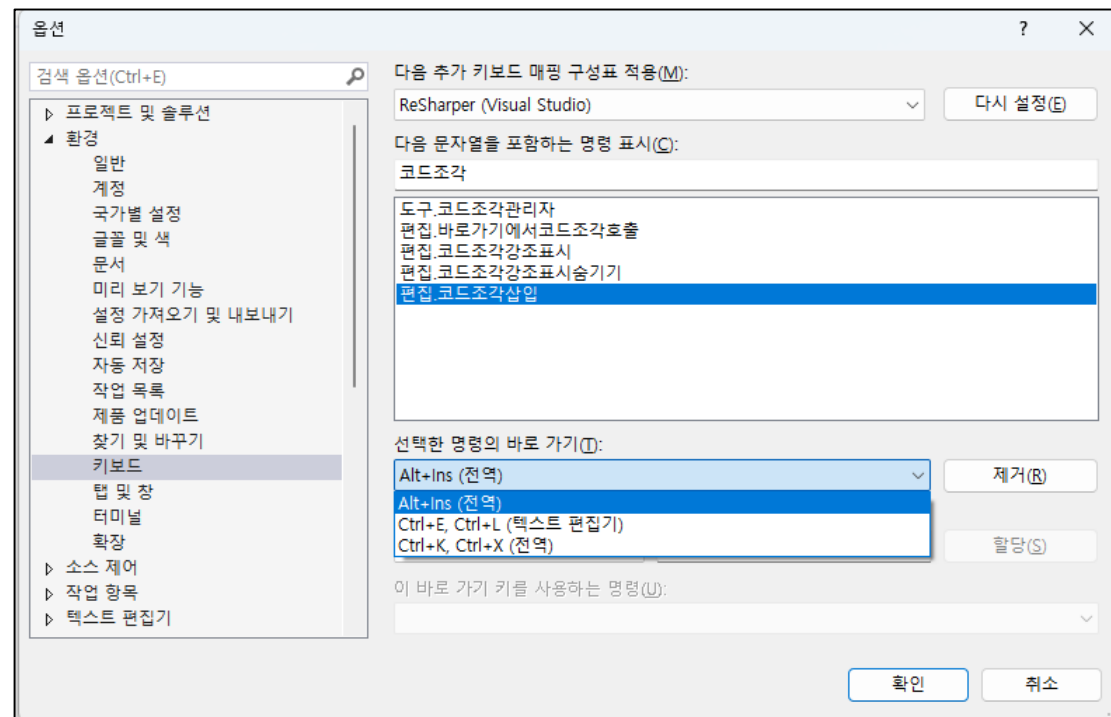
My student cannot activate a coupon code. What could be wrong?

수동 설정 1

✓도구 > 옵션 > 환경 > 키보드

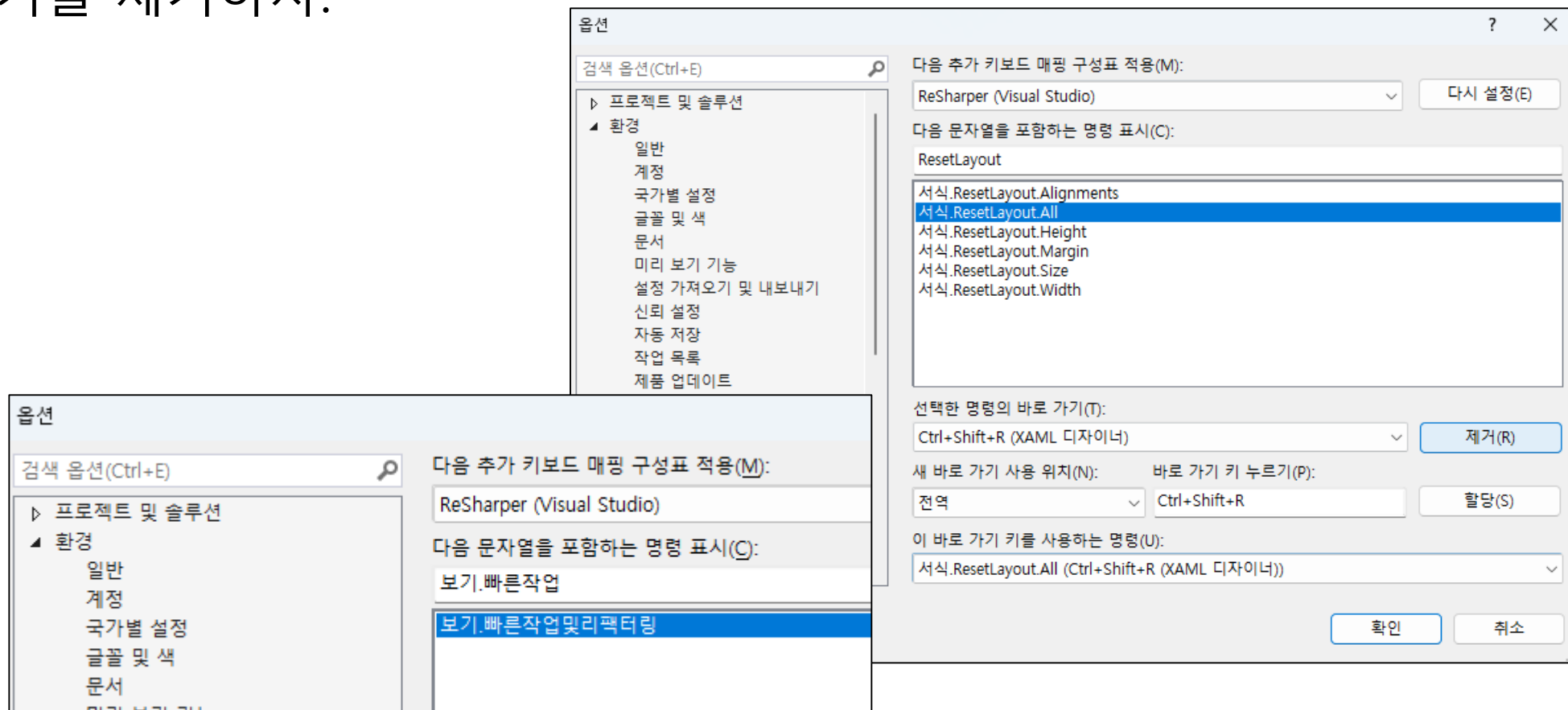
■ ReSharper 로 단축키 매핑 설정

■ “코드조각” 검색 후
Alt+Ins 단축키 “제거”
(Generate 단축키와 충돌됨)



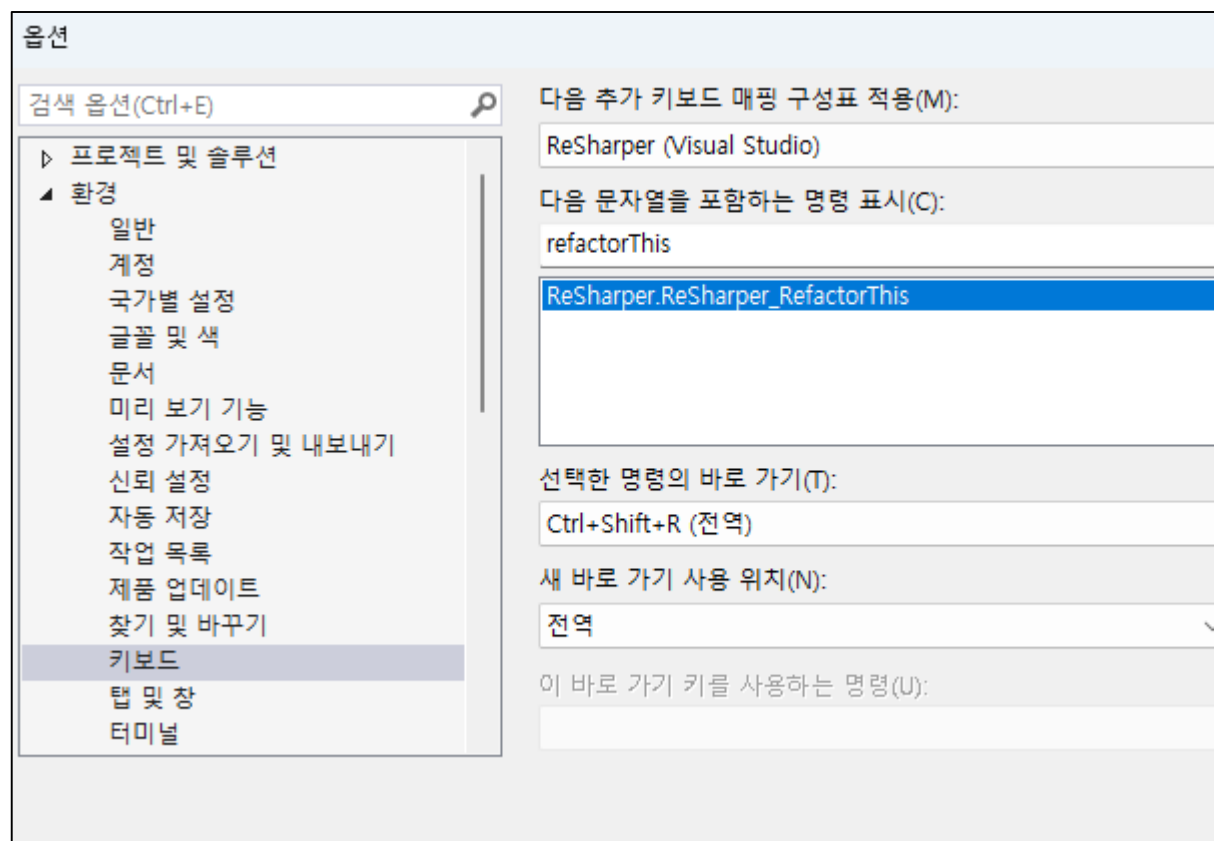
수동 설정 2

- ✓ Ctrl + Shift + R 단축키 : Refactoring This 와 충돌나는 단축키를 제거하자.



Ctrl + Shift + R 단축키 등록

✓수동으로 등록해주자.

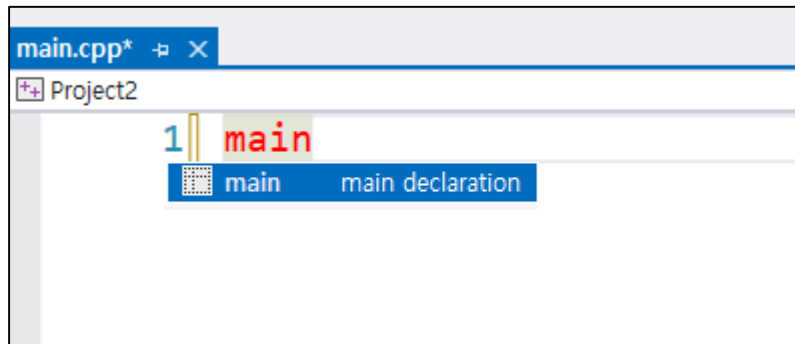


빈 소스코드에 시작

main 입력 후 Tab 누르면 자동완성

main 입력 후 ESC 한번 누르고, 다시 Ctrl + Space

- **Ctrl + Space : Symbol code completion**




```
int main(int argc, char* argv[])  
{  
    ...  
}
```

Quick Fix

✓ 오류를 고치는 방법을 제안한다.

- Alt + Enter

```
int main(int argc, char* argv[])
{
    cout
}
```


 **cout** Write to standard output stream

```
int main(int argc, char* argv[])
{
    std::cout << "HI";
}
```

main.cpp

Project2

```
1 int main(int argc, char* argv[])
2 {
3     std::cout << "HI";
4
5 }
```

 Use global variable 'std::cout' (#include <iostream>)

Ctrl + D / Shift + Delete

✓한줄 복제 / 한줄 삭제

```
int main(int argc, char* argv[])
{
    int ret = getSum(10, 20);

    std::cout << "HI";
    std::cout << "HI";
    std::cout << "HI";
    std::cout << "HI";
}
```


Alt + 드레그

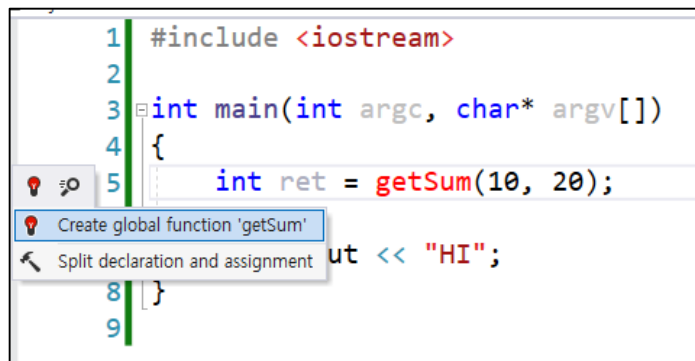
✓커서 한꺼번에 선택

```
int main(int argc, char* argv[])
{
    int ret = getSum(10, 20);

    std::cout << "HOHO HI";
    std::cout << "HOHO HI";
    std::cout << "HOHO HI";
}
```

getSum 추가 후 Alt + Enter

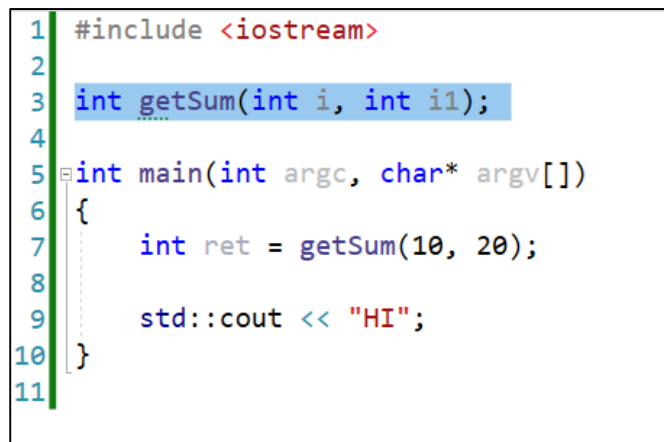
✓Create 함수 > Tab 여러 번 누르기



```
1 #include <iostream>
2
3 int main(int argc, char* argv[])
4 {
5     int ret = getSum(10, 20);
6     std::cout << "HI";
7 }
8
9
```

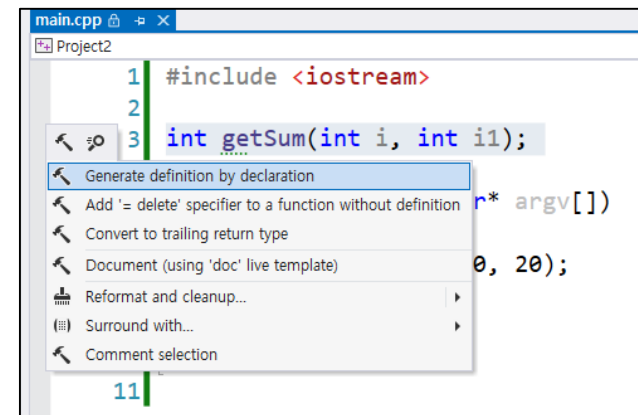
Create global function 'getSum'

Create 함수 이후 탭 여러번



```
1 #include <iostream>
2
3 int getSum(int i, int i1);
4
5 int main(int argc, char* argv[])
6 {
7     int ret = getSum(10, 20);
8     std::cout << "HI";
9 }
10
11
```

블럭잡고 다시 Alt + Enter



```
main.cpp
Project2
1 #include <iostream>
2
3 int getSum(int i, int i1);
4
5 int main(int argc, char* argv[])
6 {
7     int ret = getSum(10, 20);
8     std::cout << "HI";
9 }
10
11
```

Generate definition by declaration

선언에 대한 정의 생성하기

이름바꾸기

✓return 문 수동 입력 후, Ctrl + R, R

```
#include <iostream>

int getSum(int i, int i1)
{
    return i + i1;
}
```



```
#include <iostream>

int getSum(int t1, int t2)
{
    return t1 + t2;
}
```

fori 와

fori 입력 후 Tab 키 2회 > 10 입력 > 탭키

- cout << "HI 까지 입력 후 Ctrl + Shift + Enter 입력 (코드 완성하기)

```
#include <iostream>

int getSum(int t1, int t2)
{
    fori
    {
        # _Format_string_impl_
        # _Printf_format_string_impl_
        # _Scanf_format_string_impl_
        # _Scanf_s_format_string_impl_
    }
}
```

```
int getSum(int t1, int t2)
{
    for (int i = 0; i < 10; ++i)
    {
        cout << "HI
    }

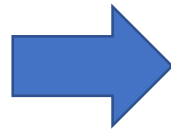
    return t1 + t2;
}
```

Ctrl + Alt + Enter

✓Reformat code

- Shift + Tab으로 들여쓰기를 지저분하게 만든 후, 테스트 해보기

```
1 #include <iostream>
2
3 int getSum(int t1, int t2)
4 {
5     for (int i = 0; i < 10; ++i)
6     {
7         std::cout << "HI";
8     }
9
10
11     return t1 + t2;
12 }
13
14 int main(int argc, char* argv[])
15 {
16     int ret = getSum(10, 20);
17
18     std::cout << "HI";
19 }
20
```



```
#include <iostream>

int getSum(int t1, int t2)
{
    for (int i = 0; i < 10; ++i)
    {
        std::cout << "HI";
    }

    return t1 + t2;
}

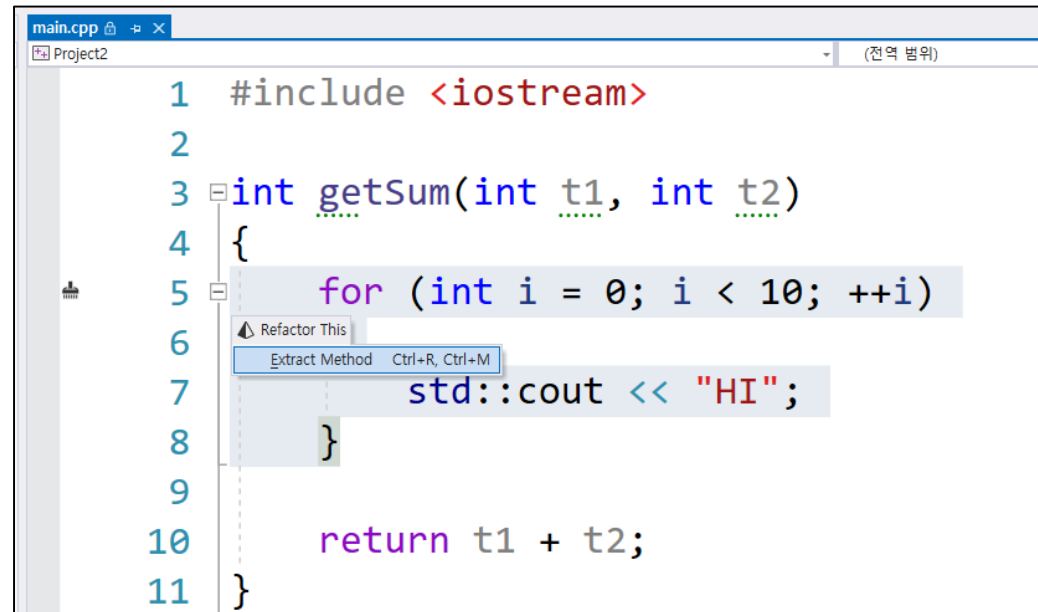
int main(int argc, char* argv[])
{
    int ret = getSum(10, 20);

    std::cout << "HI";
}
```

Refactor This

✓Ctrl + Shift + R

- Refactor This가 뜨지 않는다면, 다른 단축키와 충돌 나고 있는 것이다.
- 수동으로 단축키 재설정 해야한다.



The screenshot shows a code editor window titled 'main.cpp' with a 'Project2' tab. The code is as follows:

```
1 #include <iostream>
2
3 int getSum(int t1, int t2)
4 {
5     for (int i = 0; i < 10; ++i)
6     {
7         std::cout << "HI";
8     }
9
10    return t1 + t2;
11 }
```

A tooltip is visible over the code block from line 5 to line 8. It contains the text 'Refactor This' and 'Extract Method Ctrl+R, Ctrl+M'.

```
#include <iostream>

void do_work()
{
    for (int i = 0; i < 10; ++i)
    {
        std::cout << "HI";
    }
}

int getSum(int t1, int t2)
{
    do_work();

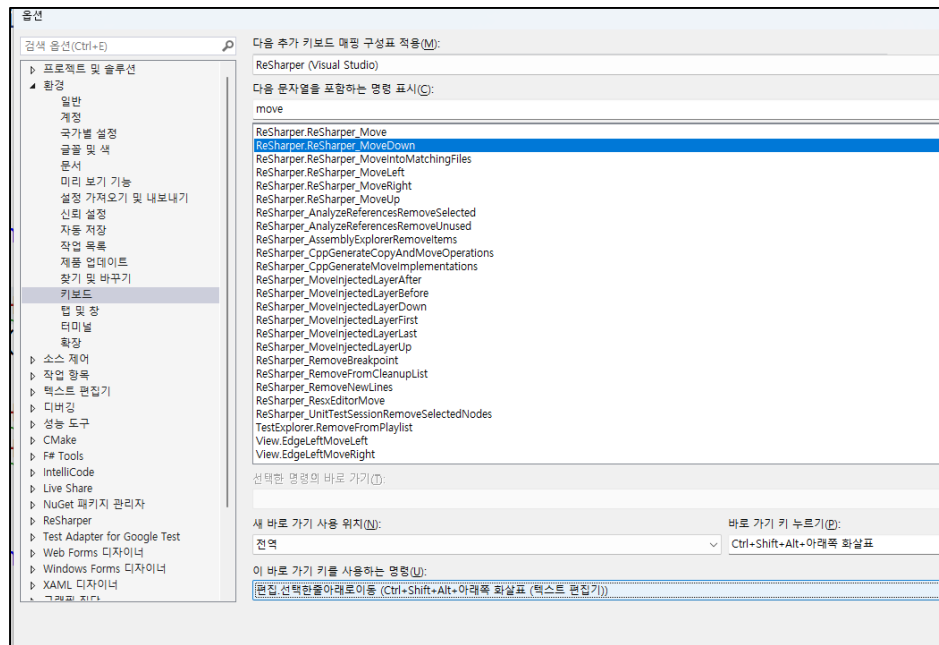
    return t1 + t2;
}

int main(int argc, char* argv[])
{
    int ret = getSum(10, 20);

    std::cout << "HI";
}
```

소스코드 이동시키기

- ✓블록잡고
Ctrl + Alt + Shift + UP / Down

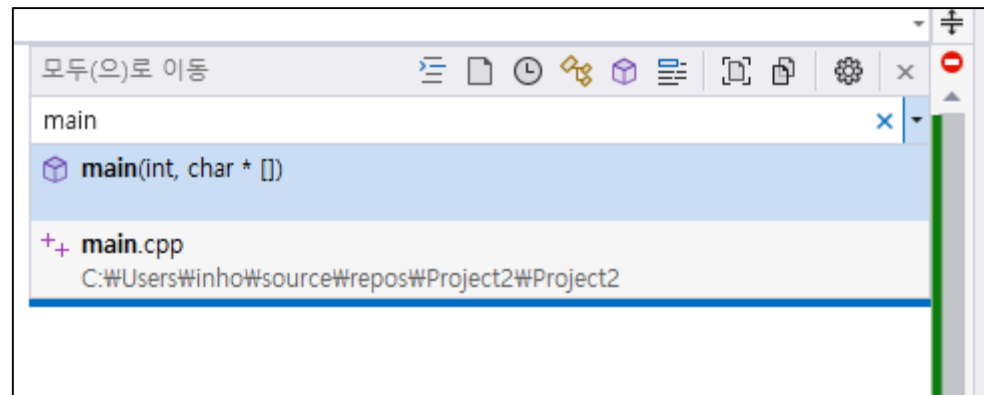


Visual Studio 의
코드 이동기능을 ReShaper 기능으로 변경

```
11 int main(int argc, char* argv[])
12 {
13     int ret = getSum(10, 20);
14
15     std::cout << "HOHO HI";
16     std::cout << "HOHO HI";
17     std::cout << "HOHO HI";
18 }
19 int getSum(int t1, int t2)
20 {
21
22
23     return t1 + t2;
24 }
25
```

파일 찾기

✓빠르게 파일을 열 수 있음



Create

Symbol code completion	✓ Ctrl+Space
Smart code completion	Ctrl+Alt+Space
Parameter info	Ctrl+Shift+Space
Go to previous signature (in parameter info)	Ctrl+Shift+Alt+Space
Extend selection	Ctrl+Alt+Right
Shrink selection	Ctrl+Alt+Left
Duplicate a line or selection	✓ Ctrl+D
Select containing declaration	Ctrl+Shift+[
Comment/uncomment line	Ctrl+Alt+/ Ctrl+Shift+/ Ctrl+Shift+`
Comment/uncomment block	Ctrl+Shift+`
Generate code (constructors, properties, implementing/overriding members, etc)	✓ Alt+Ins
Insert live template	Ctrl+E, L
Surround with template	Ctrl+E, U
Create file from template	Alt+Ins in Solution Explorer
Go to last edit location	Ctrl+Shift+Backspace
View recent files	Ctrl+,
View recent edits	Ctrl+Shift+,
Go to related files	Ctrl+Alt+F7
View bookmarks	Ctrl+`
Go to bookmark	Ctrl+[numeric key]
Set/remove bookmark	Ctrl+Shift+[numeric key]
Quick documentation	Ctrl+Shift+F1

Explore

Go to everything	✓ Ctrl+T
Go to type	Ctrl+T (second hit)
Go to symbol	Shift+Alt+T
Go to word	Ctrl+T+T+T
Go to file	Ctrl+Shift+T
Go to file member	Alt+W
Find usages	Shift+F12
Find usages (advanced)	Ctrl+Shift+Alt+F12
Find Results window	Ctrl+Alt+F12
Highlight usages in file	Shift+Alt+F11
Go to previous usage	Ctrl+Alt+PgUp
Go to next usage	Ctrl+Alt+PgDn
Navigate to	Alt+`
Go to declaration/definition	F12
Go to type of symbol	Ctrl+Shift+F11
Go to implementation	Ctrl+F12
Go to base symbols	Alt+Home
Go to derived symbols	Alt+End
Go to usage	Shift+Alt+F12
File structure	Ctrl+Alt+F
Go to next/previous member	Alt+ ↓/↑
Go to containing declaration	Ctrl+[
To-do items	Ctrl+Alt+D
Locate in Solution Explorer	Shift+Alt+L

Improve

Show available quick-fixes and context actions	✓ Alt+Enter
Inspect this	Ctrl+Shift+Alt+A
Inspection Results window	Ctrl+Alt+V
View type hierarchy	Ctrl+E, H
Go to next highlight (error, warning or suggestion)	Alt+PgDn
Go to previous highlight (error, warning or suggestion)	Alt+PgUp
Refactor this	✓ Ctrl+Shift+R
Rename	✓ Ctrl+R, R
Extract method	Ctrl+R, M
Introduce variable	Ctrl+R, V
Change signature	Ctrl+R, S
Code cleanup	Ctrl+E, C
Apply code style	Ctrl+Alt+S
Reformat code	✓ Ctrl+Alt+Enter

✓ 앞으로 수업에 자주 사용할 단축키