

**2023.2**

**Principle of Programming Language**

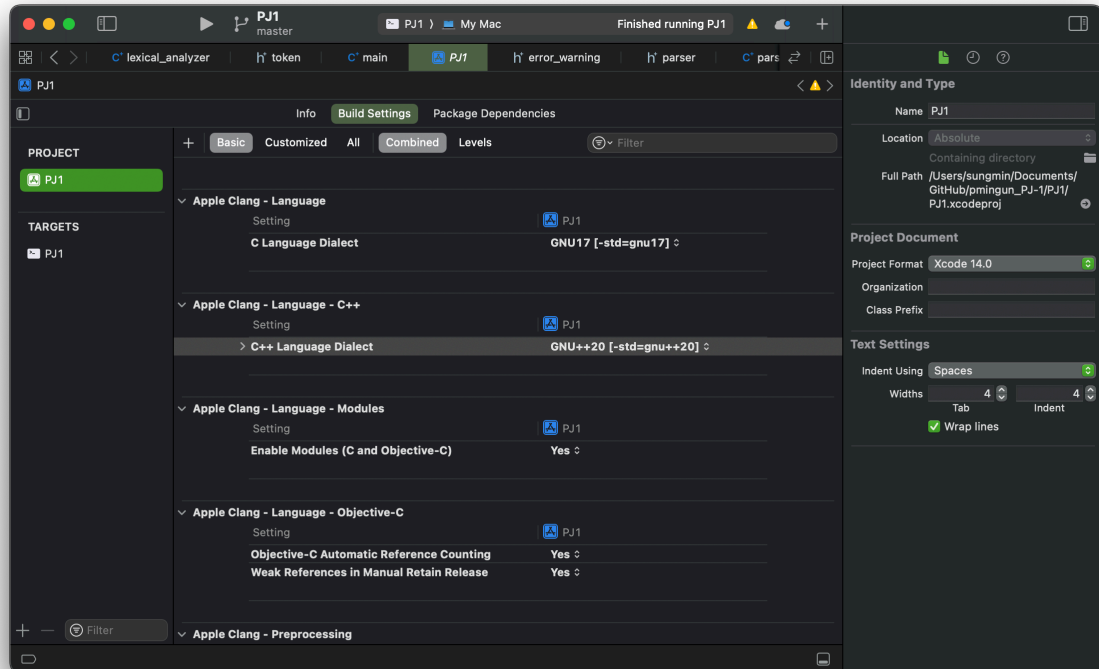
**Programming Assignment #1**

**External Documentation**

20184256	박성민
20226041	김규리

## XCode 환경에서의 실행

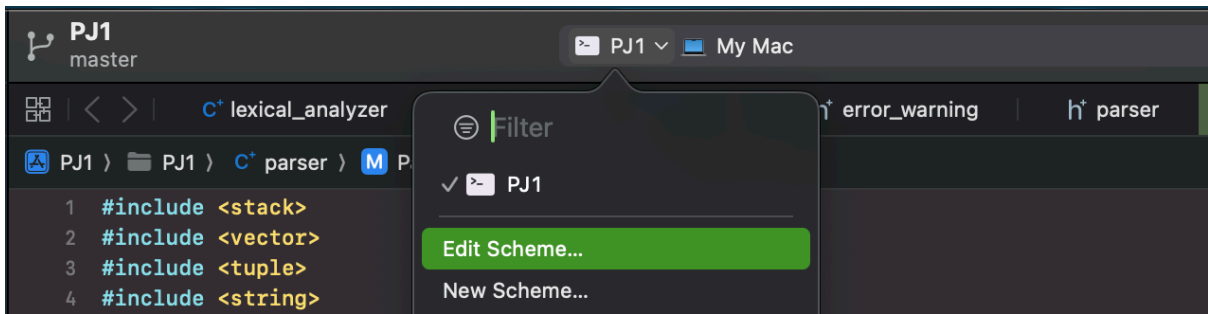
본 프로젝트는 MacOS 환경에서는 XCode 14.0, C++ GNU++20 의 환경에서 진행되었다.



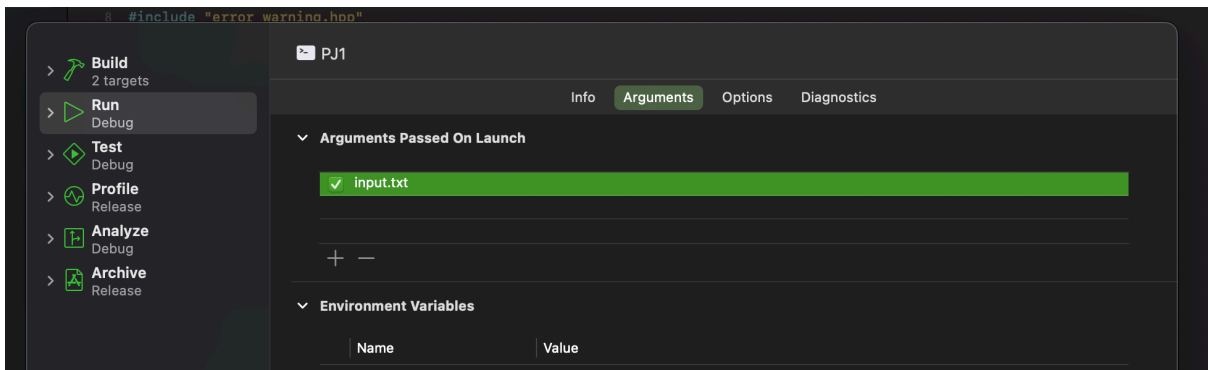
첨부된 파일 중 xcodeproj 확장자의 파일을 열어 XCode 프로그램을 실행시킨다.

이름	수정일	크기	종류
> .vscode	2023년 11월 2일 오전 9:56	--	폴더
> PJ1	오늘 오후 11:06	--	폴더
PJ1.vsproj.sln	2023년 11월 1일 오후 2:49	1KB	문서
PJ1.vsproj.vcxproj	2023년 11월 2일 오후 2:59	6KB	문서
PJ1.vsproj.vcxproj.filters	2023년 11월 2일 오후 2:59	2KB	문서
PJ1.xcodeproj	2023년 11월 3일 오전 1:47	57KB	Xcode Project

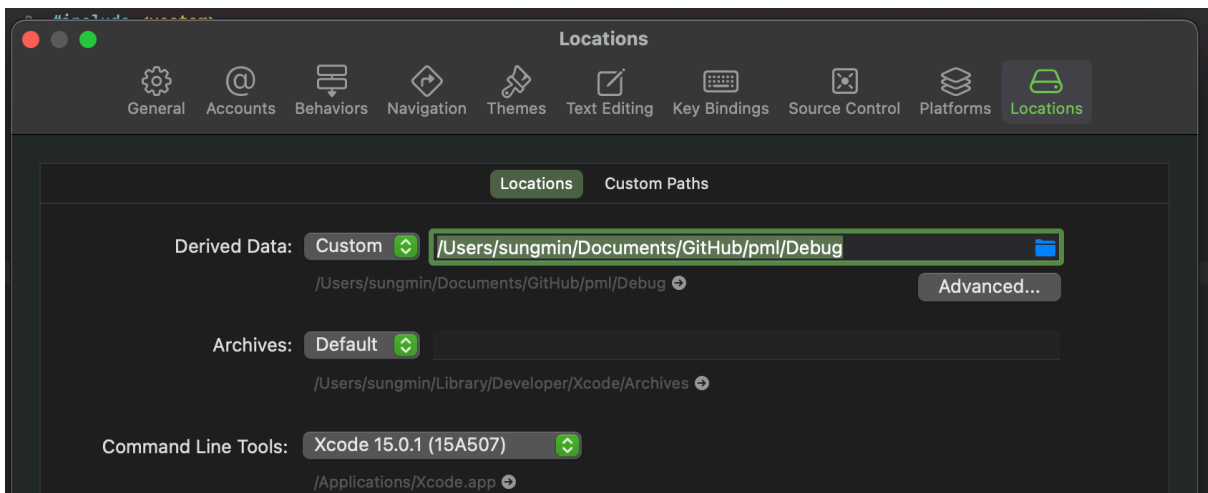
이후, XCode 화면 상단의 스키마 설정을 누른 뒤, Edit Scheme 에 진입한다.



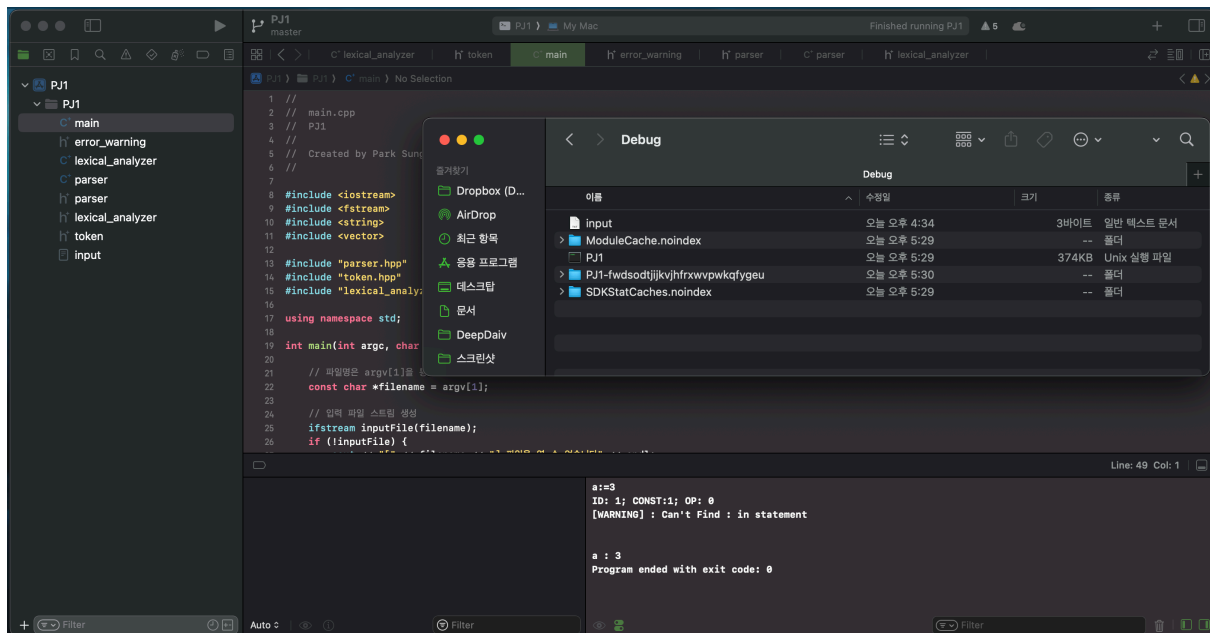
Edit Scheme 의 좌측 Run 메뉴에서 Arguments Passed On Launch 에 실행 시 main 함수에게 인자로 전달하고자 하는 파일명을 추가해준다. 아래 예시의 경우 input.txt 를 넣어주었다.



프로젝트 설정 (cmd + , ) 에서 가장 우측에 있는 Locations 메뉴에 진입해, Derived Data 의 경로를 Custom 으로 변경 후 임의의 폴더로 선택해준다. 해당 위치가 프로그램 실행 시 input 파일을 읽으려고 시도할 위치이다.



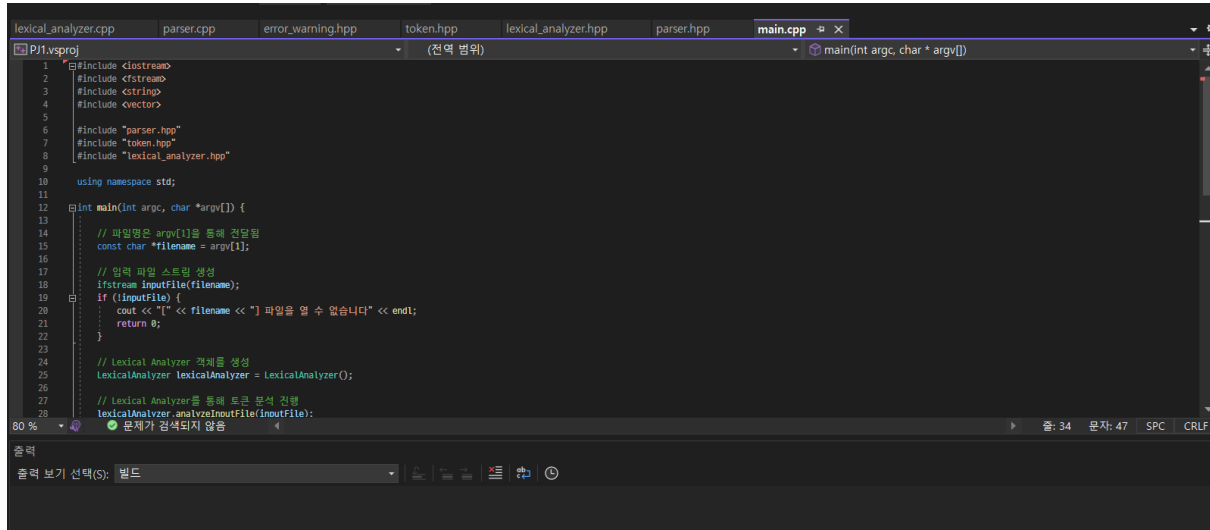
선택해준 폴더 위치에 입력 파일을 놓고 Run (cmd + R) 을 실행하면 임시 파일이 생성되며 프로그램이 정상적으로 실행되며, 콘솔에 프로그램의 로그가 출력되는 것을 확인할 수 있다.



## Visual Studio 환경에서의 실행

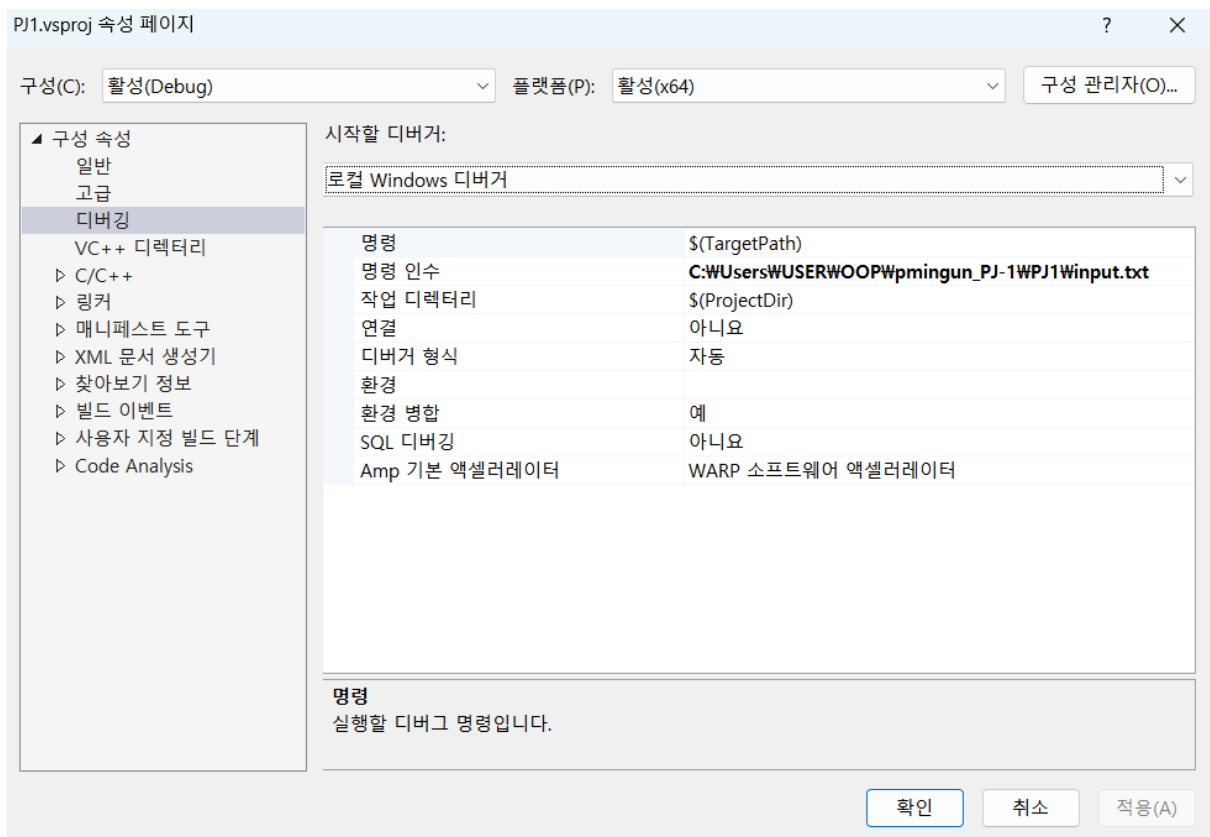
본 프로젝트는 Window 환경에서는 Visual Studio 2022, C++ 14 의 환경에서 진행되었다.

첨부된 파일 중 sln 확장자의 파일을 열어 Visual Studio 프로그램을 실행시킨다.

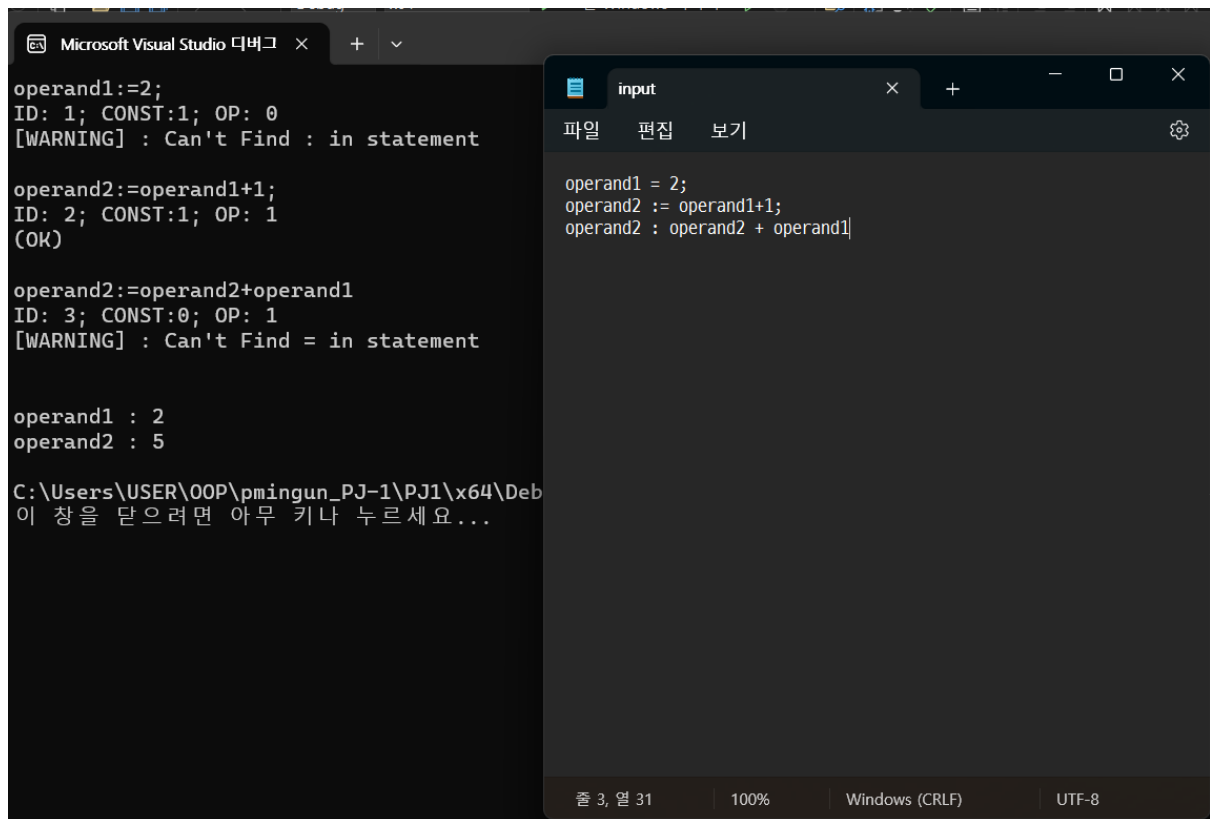


```
1 #include <iostream>
2 #include <fstream>
3 #include <string>
4 #include <vector>
5
6 #include "parser.hpp"
7 #include "token.hpp"
8 #include "lexical_analyzer.hpp"
9
10 using namespace std;
11
12 int main(int argc, char *argv[]) {
13     // 파일명은 argv[1]을 통해 전달됨
14     const char *filename = argv[1];
15
16     // 입력 파일 스트림 생성
17     ifstream inputFile(filename);
18     if (!inputFile) {
19         cout << "[*] << filename << "] 파일을 열 수 없습니다" << endl;
20         return 0;
21     }
22
23     // Lexical Analyzer 객체 생성
24     LexicalAnalyzer lexicalAnalyzer = LexicalAnalyzer();
25
26     // Lexical Analyzer를 통해 토큰 분석 진행
27     lexicalAnalyzer.analyze(inputFile);
28 }
```

프로젝트 -> 프로젝트 속성으로 이동하여 명령인수를 추가한다. 아래 input.txt 예시에서처럼 input file 로 사용할 txt 파일의 경로를 인수로 작성하면 원하는 파일을 프로그램에 전달할 수 있다.



컴파일 후 실행(Ctrl+F5)으로 실행한다.



The screenshot shows the Microsoft Visual Studio interface. On the left, the '디버그' (Debug) window is open, displaying the following output:

```
operand1:=2;  
ID: 1; CONST:1; OP: 0  
[WARNING] : Can't Find : in statement  
  
operand2:=operand1+1;  
ID: 2; CONST:1; OP: 1  
(OK)  
  
operand2:=operand2+operand1  
ID: 3; CONST:0; OP: 1  
[WARNING] : Can't Find = in statement  
  
operand1 : 2  
operand2 : 5  
  
C:\Users\USER\OOP\pmingun_PJ-1\PJ1\x64\Deb  
이 창을 닫으려면 아무 키나 누르세요...
```

On the right, the 'input' code editor is open, showing the following code:

```
operand1 = 2;  
operand2 := operand1+1;  
operand2 : operand2 + operand1
```

The status bar at the bottom indicates '줄 3, 열 31' (Line 3, Column 31), '100%' zoom, 'Windows (CRLF)' line endings, and 'UTF-8' encoding.