

Object-Oriented Programming

Lab #01

Department: 응용물리학과

Student ID: 2017103038

Name: 권인회

1. What is a compiler?

소스코드를 기계어 코드로 번역해주는 프로그램이다.

2. How is compiled code different from source code?

소스코드는 사용자가 작성하는 코드이고 컴파일된 코드는 컴퓨터가 직접 내용을 이해하고 처리할 수 있는 기계어 코드이다.

3. What tool does a programmer use to produce C++ source code?

IDEs (Integrated Development Environments)를 사용하여 소스 코드를 생성한다.

4. What tool(s) does a programmer use to convert C++ source code into executable machine code?

컴파일러를 사용하여 소스코드를 기계 코드로 변환한다.

5. What does the linker do?

컴파일 완료 후 만들어진 오브젝트 파일들을 상호 연결시켜준다.

6. Does the linker deal with files containing source code or machine language code?

Yes, linker는 소스코드를 컴파일 할 때 생기는 오브젝트 파일을 묶어서 실행할 수 있는 exe 파일을 만들어준다.

7. What does the preprocessor do to source code?

전처리 과정에서 소스 파일에 다른 파일의 텍스트를 포함시키거나 일부 문장을 다른 문장으로 바꾸는 작업 등을 수행한다.

8. List several advantages developing software in a higher-level language has over developing software in machine language.

Higher-level language는 사용자 위주의 언어기 때문에 machine language보다 사용자가 다루기 쉽고 가독성이 높다. 특정 컴퓨터와 관계없이 독립적으로 프로그램을 만들 수 있다. 대부분의 프로그래밍 언어들이 속해 배우기 쉽다. 용량이 커 큰 프로그램을 만들 수 있다. 프로그램을 짤 때 시간이 적게 걸리고 디버그도 쉽다. 등이 있다.

9. How can an IDE improve a programmer's productivity?

소스 코드 편집기를 제공하면서 시각적 신호를 활용한 구문 강조와 같은 기능을 포함하여 소프트웨어 코드를 작성하도록 돕는다. 언어별 자동 완성 기능과 코드 작성 중 버그 검사 기능도 제공하며 개발자에게 편리한 프로그래밍을 할 수 있도록 돕는다.

10. Name a popular C++ IDE is used by programmers developing for Microsoft Windows.

Microsoft Visual Studio 이다.

11. What are the primary differences between symbolic and high-level languages?

Symbolic and high-level languages는 사용자의 접근성의 차이로 구분된다. 사용자에게 친숙한 언어로 이루어질수록 **high-level languages**에 속하고 컴퓨터가 이해하기 쉬운 언어일수록 **Symbolic languages**로 분류된다.

12. What is the difference between a source code and an object module?

소스 코드는 사용자가 작성한 코드를 말하고, 오브젝트 모듈은 이 소스 코드가 컴파일 되어 만들어진 오브젝트 코드의 모듈을 말한다.

13. What preprocessor directive is necessary to use statements with the `std::cout` printing stream object?

#include <iostream> 이 필요하다.

14. What statement allows the short name `cout` to be used instead of `std::cout`?

using namespace std; 이다.

15. What does the name `std` stand for?

standard(표준)의 약자이다.

16. All C++ programs must have a function named what?

main이라는 이름의 함수가 있어야한다.

17. The body of `main` is enclosed within what symbols?

중괄호인 {}로 둘러싸여 있다.

18. What operator directs information to the `std::cout` output stream?

Insertion operator인 << 이다.

19. Write a C++ program that prints your name in the console window.

Code:

#include <iostream>

```
int main() {
    std::cout << "권인회";
}
```

Run:

권인회

C:\Users\Administrator\source\repos\WEx001\Debug\WEx001.exe(프로세스 40708개)
이(가

) 종료되었습니다(코드: 0개).

이 창을 닫으려면 아무 키나 누르세요...

20. Write a C++ program that prints your first and last name in the console window. Your first name should appear on one line, and your last name should appear on the next line.

Code:

```
#include <iostream>
int main() {
    std::cout << "Kwon" << std::endl;
    std::cout << "Inhoe";
}
```

Run:

Kwon

Inhoe

C:\Users\Administrator\source\repos\WEx001\Debug\WEx001.exe(프로세스 41672개)
이(가

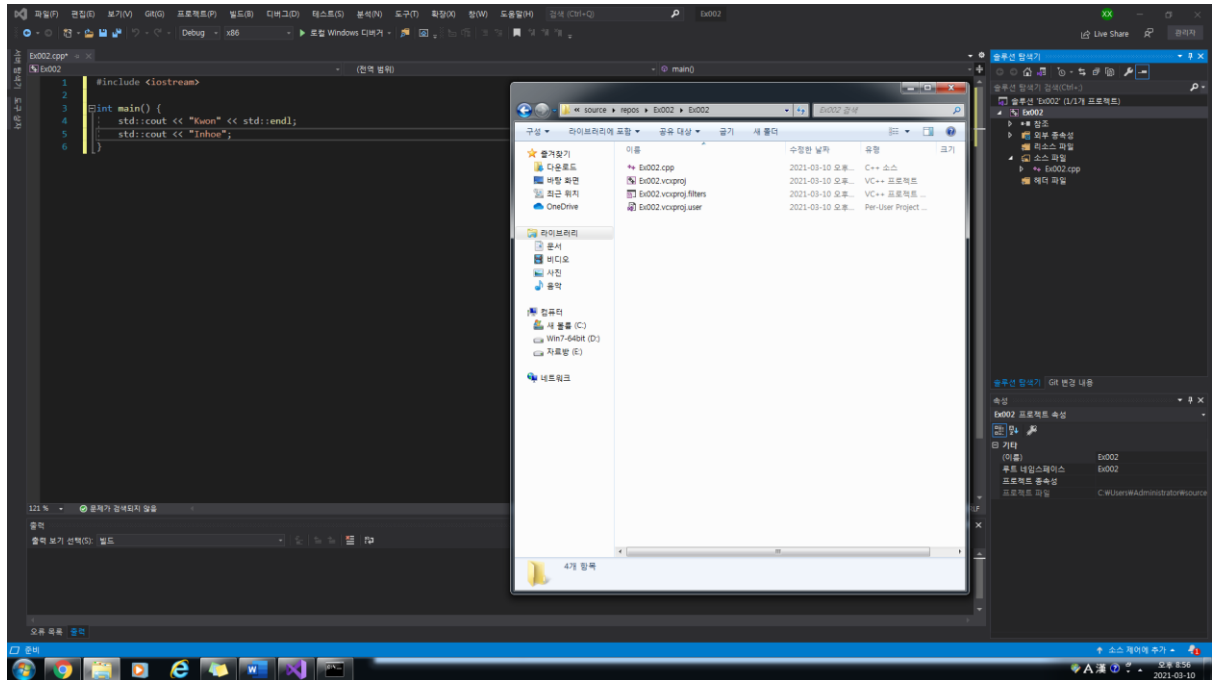
) 종료되었습니다(코드: 0개).

이 창을 닫으려면 아무 키나 누르세요...

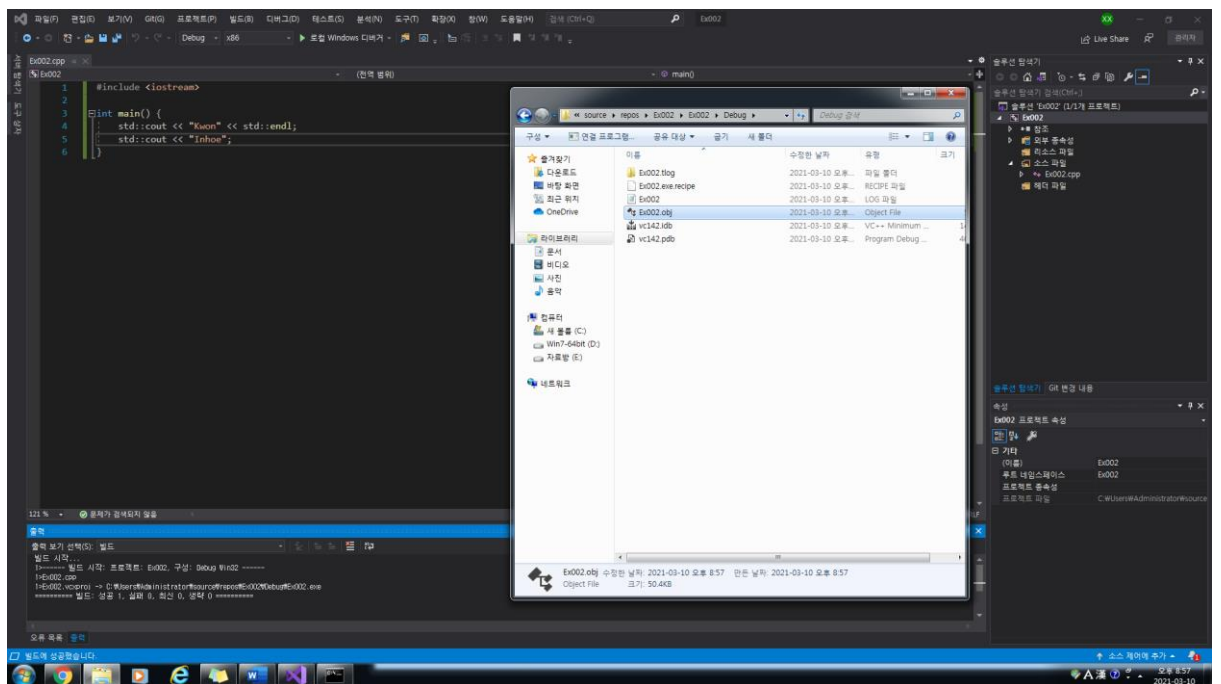
21. Write your own OS and IDE, explain the compiling and linking process, and show the screenshot.

OS : Windows 7, IDE : Microsoft Visual Studio 2019

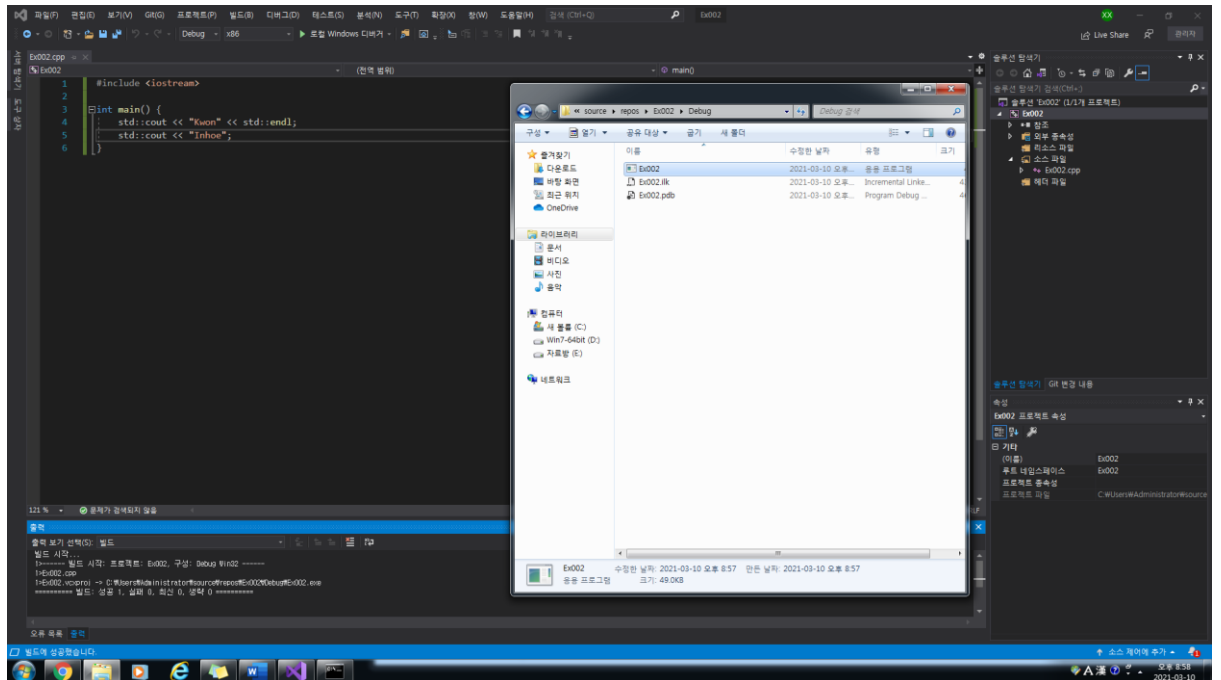
(주로 맥북을 사용하나, 아직 제 수준에서 이 수업을 들을 때 맥북 사용이 어려워 잠시 윈도우를 대여하여 사용중입니다. Mac OS에서는 Xcode를 IDE로 사용하며 이 부분에서도 강의가 진행된다면 감사히 들겠습니다.)



1. IDE를 사용하여 새 프로젝트를 콘솔 앱으로 만든 뒤, 소스 코드 작성 후 파일 경로의 상황 (아직 빌드를 하지 않음)



2. 빌드 이후 소스 코드가 컴파일된 오브젝트 파일이 생성된 것을 볼 수 있다.



3. 빌드 이후 링크 과정이 이루어졌고 exe 파일도 생성된 것이 확인된다.

Visual Studio 콘솔 창에서 파일 경로 설정 후 'cd Ex002.cpp'를 입력하면 수동으로 컴파일하여 Ex002.obj 파일이 생성되는 것을 볼 수 있다. 이것이 compile 과정이다. 만약 cpp 파일이 여러 개라면 컴파일한 소스 코드를 link하기 위해 'link Ex002.obj Ex001.obj'를 입력하면 Ex002.exe 파일이 생성되는 것을 볼 수 있다. 이것이 link과정이다.