2017년 2학기

객체지향프로그래밍

CSED232

HW1

담당교수: 윤은영

학번: 20160074

학과: 화학공학과

이름: 고진민

POVIS ID: eric9709

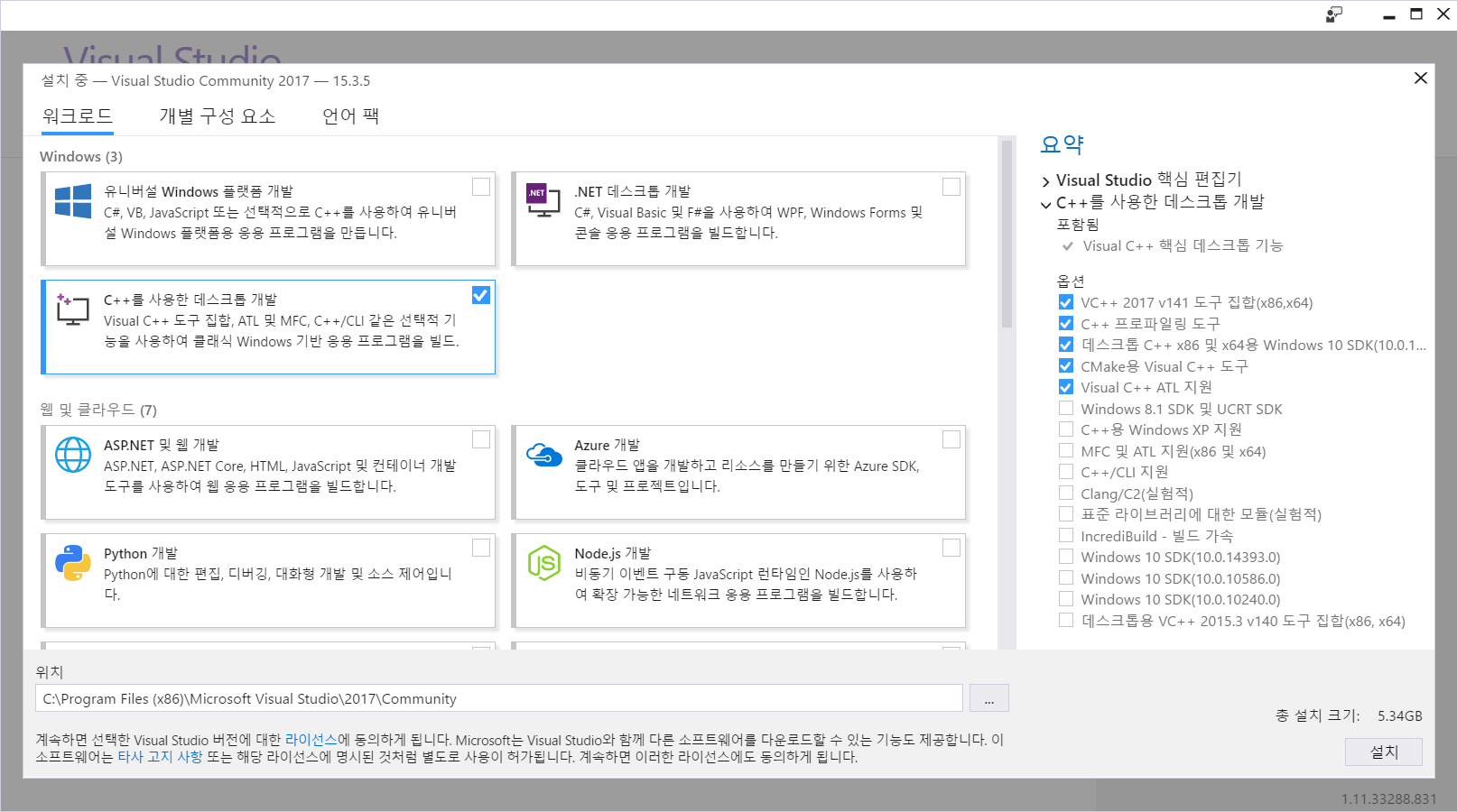
<명예서약>

**나는 이 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.**

1. **MinGW, VS 설치방법**

* VS 설치방법

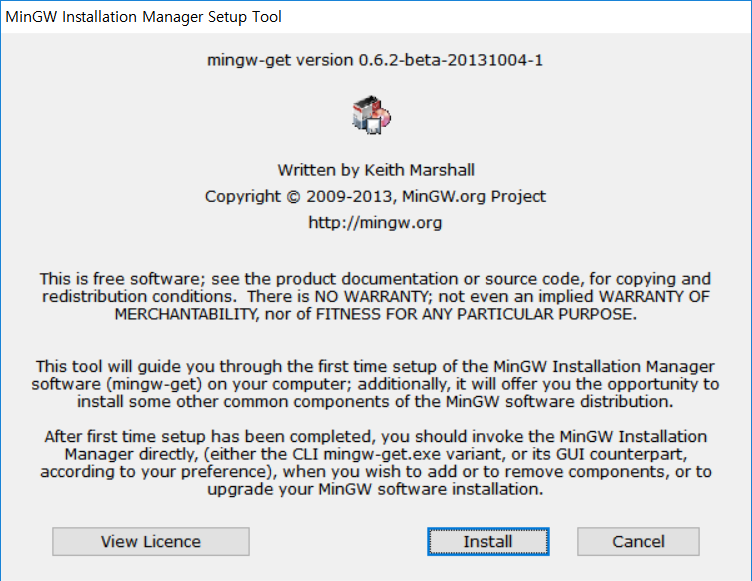
먼저, <https://www.visualstudio.com/ko/vs/community/>에 접속해 프로그램을 다운로드한다. 이후 설치 프로그램을 실행한다. 프로그램을 실행하면 아래와 같은 선택창이 뜨는데, **C++을 사용한 데스크탑 개발**을 선택하고 설치를 진행한다.



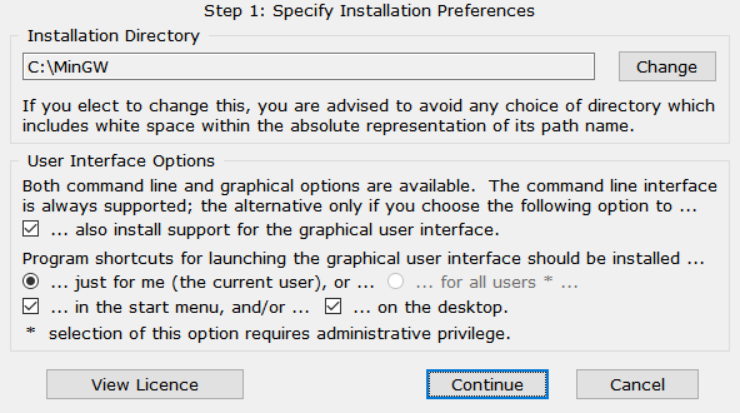
설치를 완료하면 컴퓨터를 재시작하라는 문구가 뜰 것이다. 작업 중인 파일을 저장하고 재부팅을 하자. 재부팅 후, 로그인 여부를 선택하고(학교 계정으로 로그인해도 무방) 개발 설정을 Visual C++로 설정한 뒤 프로그래밍을 진행한다. 내 경우는 재부팅 후 실행했더니 자동으로 로그인이 되어 있어서 로그인 이후부터 새로운 프로젝트 생성까지의 과정이 모두 생략되어 캡쳐를 할 수 없었다.

* MinGW 설치 및 코딩방법

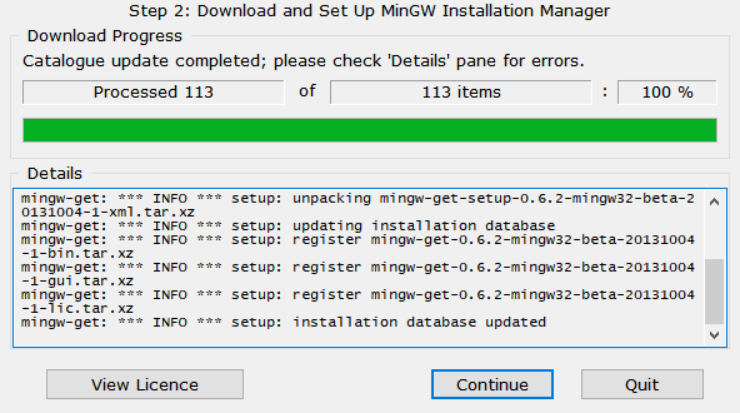
먼저, <https://sourceforge.net/projects/mingw/files/latest/download?source=files>에 접속해 MinGW 설치파일을 다운로드 받는다. 파일을 실행하면 아래 캡쳐화면과 같은 창이 뜰 것이다.



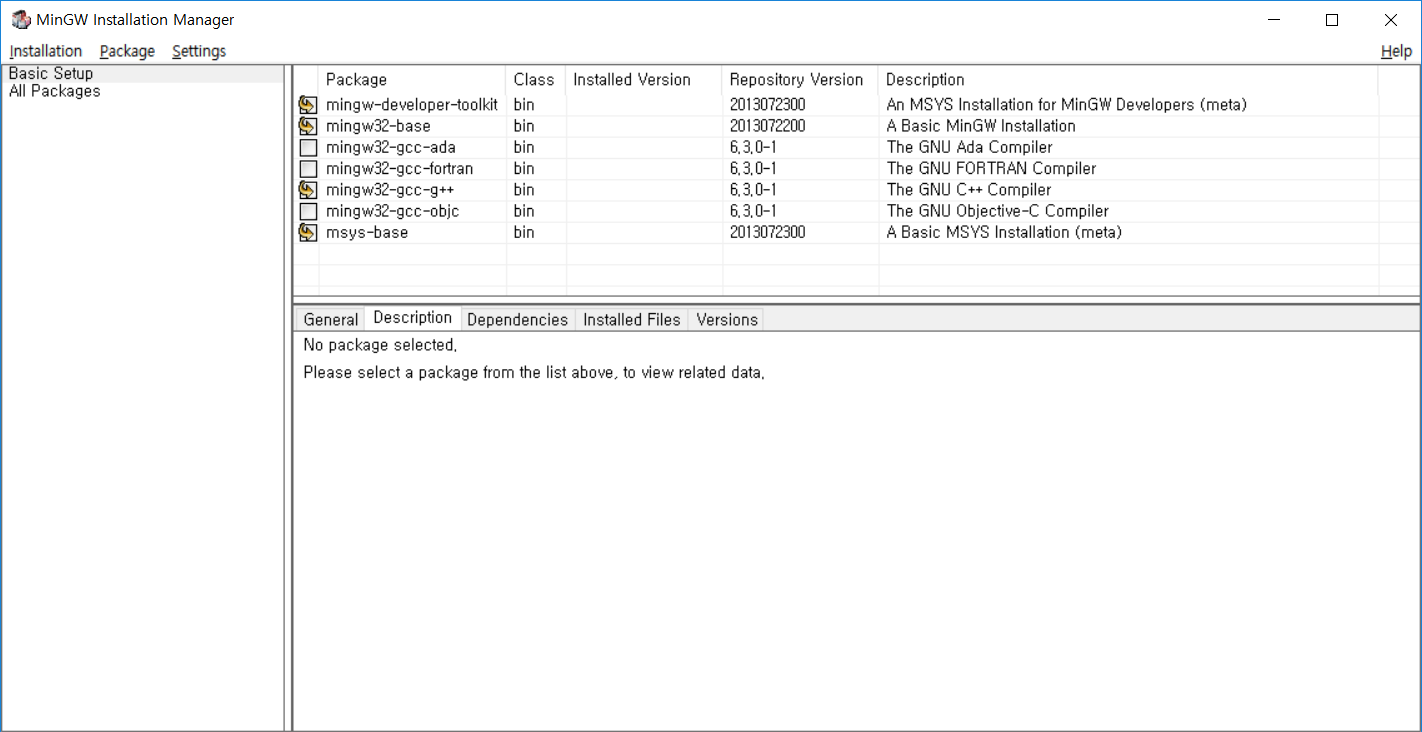
Install 버튼을 클릭하면 아래 캡쳐와 같은 화면이 뜰 것이다. 설치 경로와 GUI 설치 옵션을 선택한 뒤 계속한다.



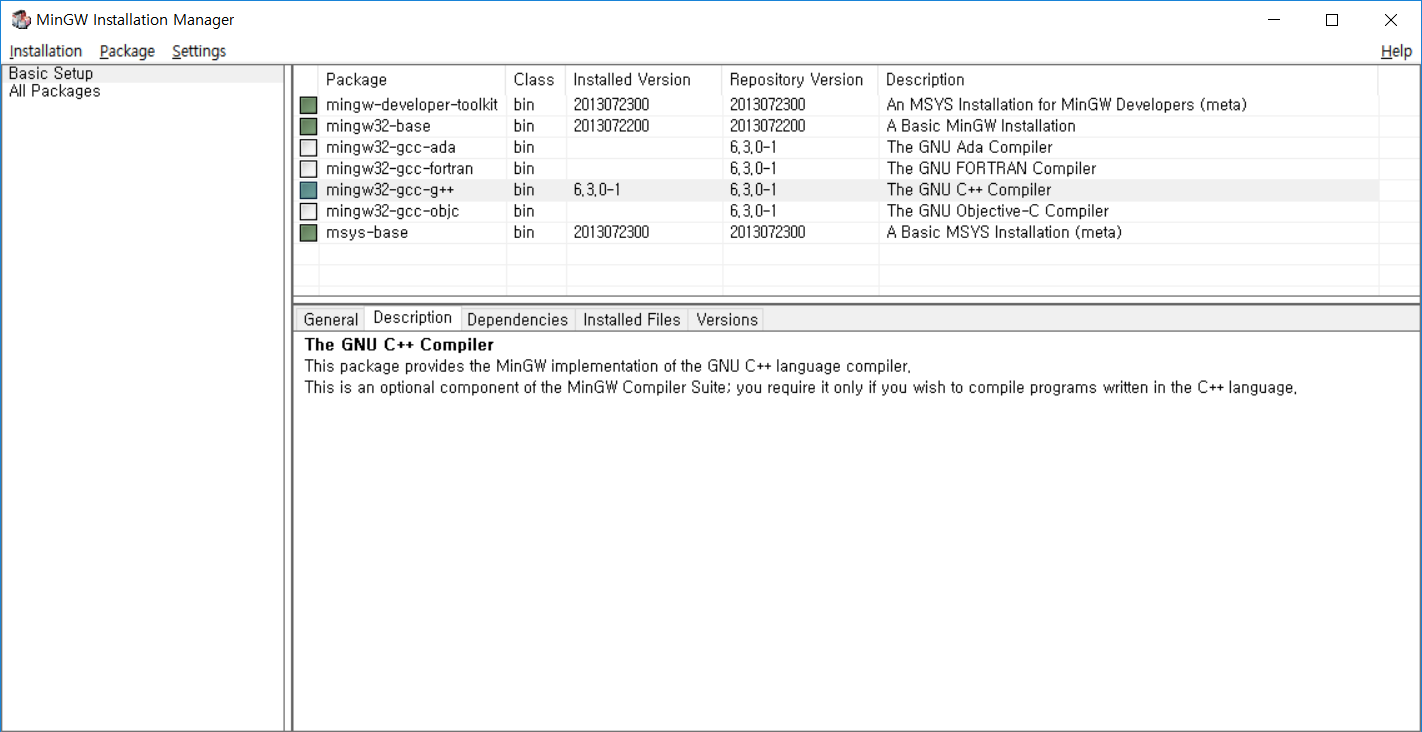
이후 다운로드가 진행될 것이고, 완료되면 아래 캡쳐와 같은 창이 뜨게 된다.



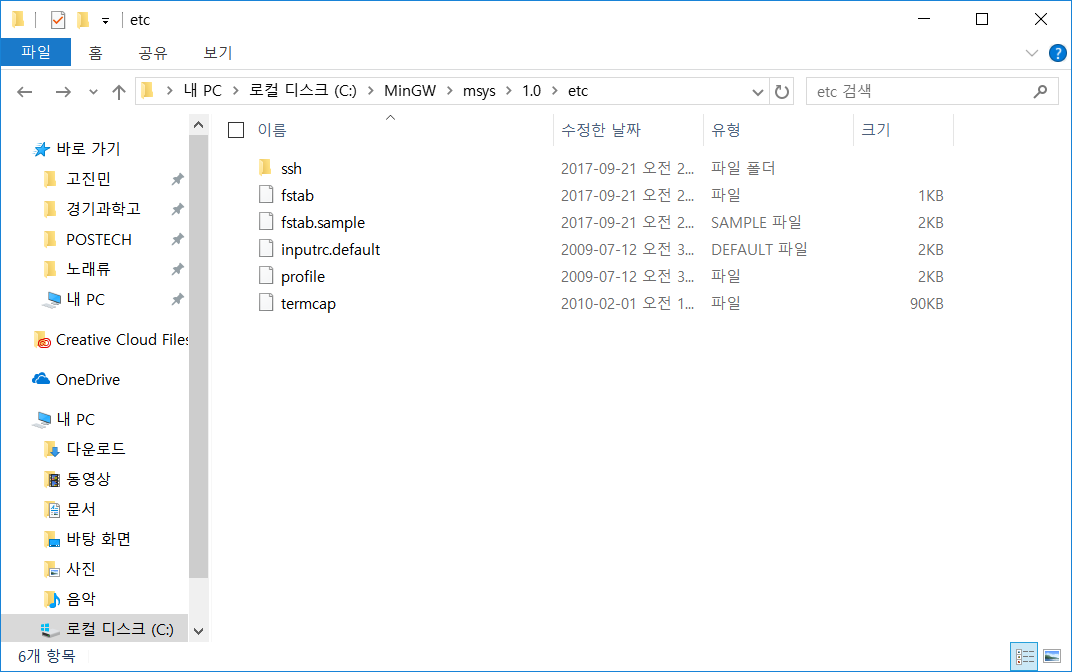
Continue를 클릭하면 아래와 같은 설치 패키지 선택 창이 뜨게 된다. 아래와 같이 선택한 후 죄측 상단 메뉴의 Installation – Apply Changes 버튼을 클릭하고 Apply 버튼을 클릭해 다운로드 및 설치를 진행한다.



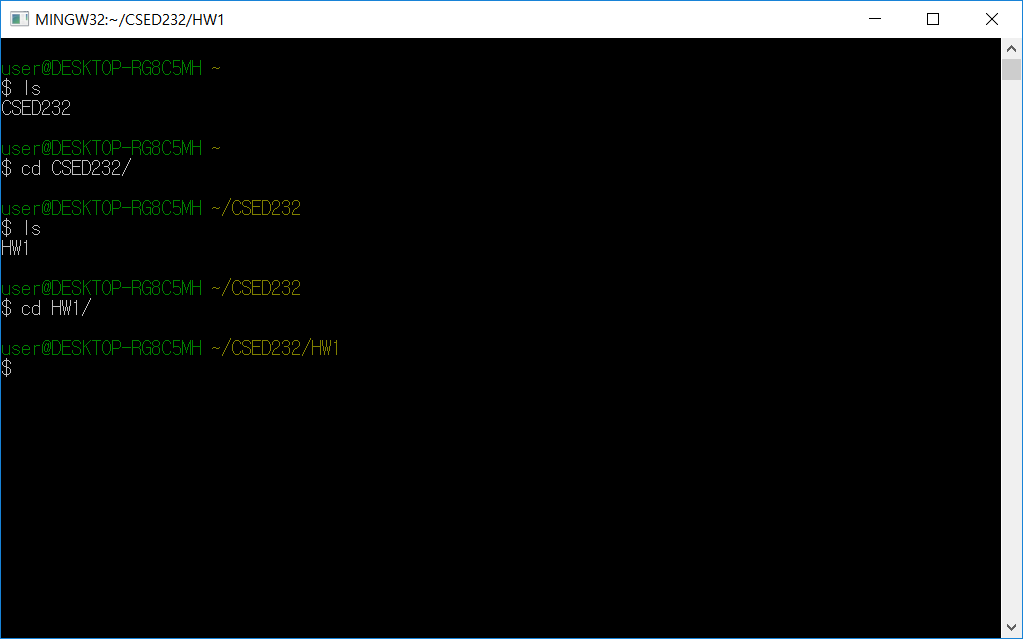
설치를 완료하면 아래 캡쳐와 같이 체크 부분이 바뀌게 된다.



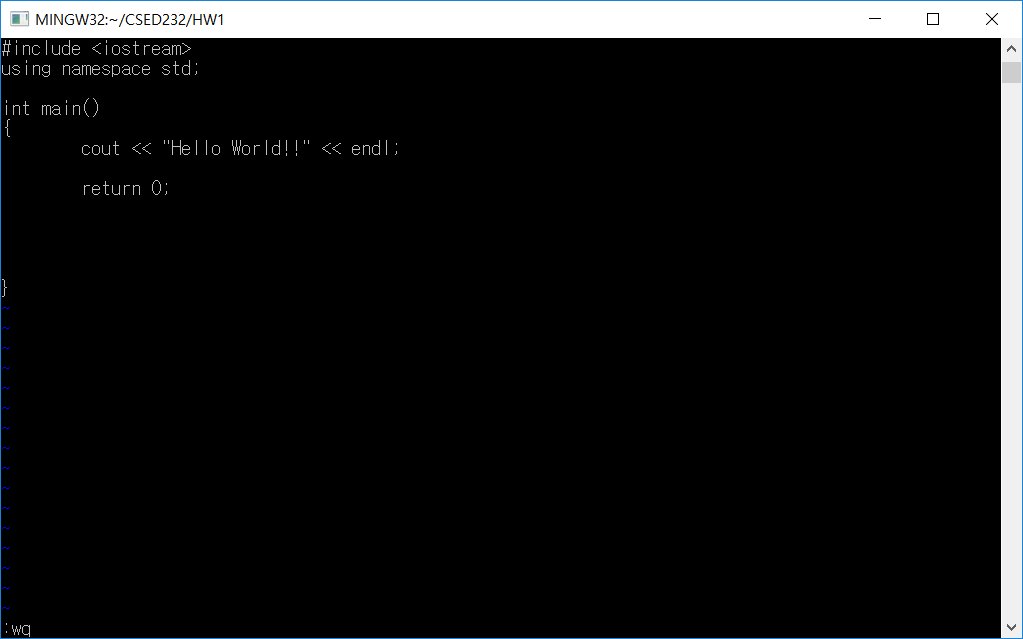
이후 설치 경로를 따라 들어가서 (설치경로)\msys\1.0\etc 에 접근한다. 이 때 fstab 파일이 존재하면 아무 일도 안 해도 되지만 없다면 fstab.sample 파일을 fstab 파일로 바꾼다.



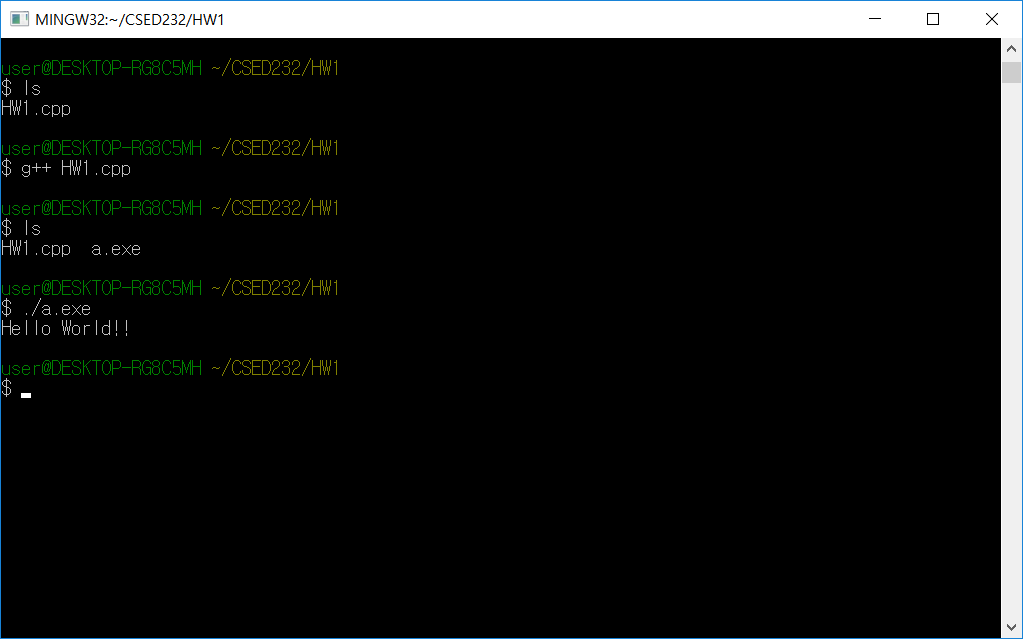
이후 (설치경로)\msys\1.0 에 있는 msys.bat 파일을 실행하면 콘솔 창이 띄워진다. 실행 준비는 끝났다. mkdir 명령어를 이용해 디렉토리를 새로 형성해주자. 나는 user/CSED232/HW1 이 되도록 디렉토리를 새로 생성했다. 이후 cd 명령어를 이용해 현재 디렉토리를 파일을 생생하길 원하는 위치로 변경한다. 아래 캡쳐는 user/CSED232/HW1 디렉토리로 이동한 모습이다.



여기서 편집할 .cpp 파일을 만들어주자. vim 명령어를 이용해 HW1.cpp 파일을 생성하자. 그러면 새로운 파일이 형성되고 코딩을 진행할 수 있다. 아래는 간단한 코드의 예시이다.



코딩을 완료했으면 :wq 명령어를 입력해 저장 후 종료한다. 이제 컴파일링을 해야 하는데 g++ 명령어를 이용해 GNU C++ 컴파일러를 사용하자. 컴파일링이 문제 없이 끝났다면 ./ 명령어를 이용해서 생성된 실행 파일을 실행하면 아래 캡쳐와 같이 나타나게 된다.



종료하고 싶을 때에는 exit 명령어로 종료하면 된다.

1. **자기소개 및 인생계획**

#include <iostream>

using namespace std;

int main()

{

cout << "저는 포스텍 화학공학과 20160074 고진민입니다. 프로그래밍 경험은 경기과학고등학교에서 배웠던 컴퓨터프로그래밍 과목을 이수한 것과 포스텍의 CSED101 과목을 이수한 경험이 있습니다.\n";

cout << "대학생 1학년은 이미 지났으므로 계획을 세울 필요가 없습니다.\n";

cout << "대학생 2학년은 2학기 계획만 생각하겠습니다. 이번 학기에는 21학점을 듣고 있고, 전부 이수하게 되면 총 이수 학점이 80학점이 됩니다. 이번 학기에 컴공과 과목으로 객체지향프로그래밍과 컴퓨터공학소개 과목을 수강하고 있습니다.\n";

cout << "대학생 3학년은 기본적으로 화공과 전필과 전선 과목을 수강하는 동시에 컴공과에서 개설되는 전필 과목을 수강할 예정입니다. 1학기에는 DS와 전필 과목을 하나 더 들을 것 같습니다. 이 과목들을 수강하면서 1학기 중에 최종적으로 컴공과 복수전공 혹은 부전공 여부를 결정할 예정입니다. 2학기에는 1학기에서 결정한 내용을 토대로 강의를 수강할 예정입니다. 방학 중에 서울대학교로 학점 교류를 생각하고 있습니다.\n";

cout << "대학생 4학년은 졸업 및 이후의 계획을 생각하면서 강의를 수강할 예정입니다. 기본적으로 군 문제 해결 방법으로 대학원에 갈 생각을 하고 있기 때문에 어느 랩에 갈지 결정할 예정입니다. 컴공과 복수전공을 하게 된다면 졸업은 5학년으로 늦춰질 거라 생각합니다.\n";

cout << "앞서 말씀드렸지만, 컴공과 복수전공 혹은 부전공을 생각하고 있습니다. 이 부분에 대해서 교수님과 면담을 하고 싶습니다. 섣부르게 판단을 내리기보다는 먼저 이번 학기에 객체지향프로그래밍과 컴퓨터공학소개 과목을 수강하면서 저에게 맞는지 확인을 해본 뒤 다음 학기에 컴공과 과목을 좀 더 수강하면서 최종적으로 결정할 예정입니다.\n";

cout << "졸업을 한 이후에는 대학원에 진학할 예정입니다. 군 문제를 해결해야 하므로 전문연구요원을 위해서는 박사 학위를 따야 할 것이라 생각합니다.\n";

cout << "대학원을 졸업한 이후에는 취업을 할 예정입니다. 교수는 염두에 두고는 있지만 진지하게 고려중인 선택지는 아닙니다. 기업에 취업해서 돈을 벌며 노후준비를 하고, 정년퇴임을 한 후에는 노후자금으로 생활할 예정입니다.\n";

cout << "결혼은 제가 데미로맨스 성향이 있기 때문에 친하게 지내다 연애를 하고 결혼을 하지 않는 이상은 할 생각이 없습니다. 자녀 계획은 배우자가 정말로 원하는 경우가 아니라면 자녀를 가지지 않을 생각입니다. 아이를 가지는 순간 부모, 특히 여성이 자신의 삶을 잃게 되고, 그와는 별개로 육아 비용도 엄청나기 때문이며 제게 한 생명의 미래를 책임질 자격이 있다고 생각하지 않기 때문입니다.\n";

return 0;

}

1. **용어 설명**
2. Procedural Programming / Object-Oriented Programming

Procedural Programming은 절차적 혹은 절차지향적 프로그래밍이라는 의미이다. 간단히 말하면 수행될 명령들이 연속적으로 나열되어 있는 프로그래밍을 의미한다. 순차적인 처리가 중요시되며 코드 전체가 유기적으로 연결되어 있기 때문에 어느 한 부분에서 문제가 생기면 전체가 동작하지 않는다. 보통 위에서 아래로 내려가는 순서로 진행된다. 대표적으로 우리가 사용하던 언어인 C언어가 있다.

Object-Oriented Programming은 객체지향 프로그래밍이라는 의미다. 절차지향과는 다르게 프로그래밍에 있어서 프로그램을 연속적인 프로그램의 나열 혹은 목록이 아닌 특정 기능을 수행하는 객체들이 모여 서로 상호작용하는 것으로 보는 시각이다. 객체 지향은 여러 특성을 지니는데, 캡슐화(관련된 데이터와 코드가 하나의 묶음으로 정리된 것), 상속, 다형성 등이 있다. 이런 특징 덕분에 유지보수와 코드 재활용이 쉽다는 장점이 있다.

1. Abstract Data Types

Abstract Data Type, ADT는 추상적 자료형 이라고도 한다. 간단히 말해, 프로그램의 기능에 대해 구체적인 구현 부분은 나타내지 않고 순수한 기능만을 나열한 것을 의미한다. 선풍기를 예로 들면 설명서라고 생각할 수 있다. 설명서에는 어떤 버튼이 어떤 기능을 수행하는지 나타나 있지만, 내부 회로에서 무슨 일이 벌어지고 있는지는 설명하고 있지 않다. 이는 프로그램의 구현자와 사용자를 분리한다는 장점이 있다. 객체지향 언어에서는 ADT가 클래스 개념과 같다.

1. Object / Class

Object는 객체라고도 한다. 객체는 실제로 존재하는 구체적인 대상 혹은 시스템이다. 객체는 상태(Attribute)와 행위(Method)를 가진다. 특정 기능을 수행하는 하나의 단위라고 생각할 수도 있다.

Class는 Object의 집합이다. 예를 들어 사람 한 명 한 명이 Object라고 생각하면 각 사람은 상태가 있고, 어떤 행동을 한다. 이런 각각의 사람을 모아 ‘인간’ 혹은 ‘사람’이라고 하는 집합을 생각할 수 있는데, 이것이 Class이다. 동일한 유형의 Object를 표현하는 추상적인 개념이다. C++에서는 data type과 같은 의미로 생각할 수 있다.

1. Information Hiding / Encapsulation

Information Hiding은 노출되는 것을 원하지 않는 부분을 숨기는 것을 의미한다. 가령 코드에서 실제 구현 부분 중 어떤 부분을 외부에 노출하고 싶지 않을 때, 접근 속성을 private로 지정해 노출시키지 않는 방법 등이 있다. 여기서 private는 접근제어 키워드라고 하는데, private 외에도 public, protected 등이 있다.

Encapsulation(캡슐화)은 클래스를 정의하는 데에 있어서 서로 관련이 있는 데이터나 함수들을 하나로 묶는 것을 의미한다. 캡슐화 과정을 통해서 information hiding을 동시에 진행할 수 있다. 코드의 재활용성을 높일 수 있다는 장점이 있다.

1. Inheritance

Inheritance는 상속이라고도 한다. 상속은 새로운 클래스가 기존의 클래스의 자료 및 연산을 이용할 수 있게 하는 것이다. 상속이 이루어지는 클래스 간 관계를 나타내는 용어가 있다. 상속을 받는 클래스는 하위/자식 클래스라 칭하고 새로운 클래스가 상속한 기존의 클래스를 상위/부모 클래스라 칭한다. 상속을 이용하면 하위 클래스를 이용해 상황에 따라 프로그램을 수정하거나 클래스 사이의 종속 관계를 만들어 조직화할 수 있다.

1. Polymorphism

Polymorphism은 다형성이라고도 한다. 다형성은 동일 메시지에 대해 서로 다른 방법으로 응답할 수 있는 기능이라고 생각할 수 있다. 어떤 함수가 여러 객체에서 정의되었으면, 이 함수는 이름은 같지만 값이 입력된 후 각 객체에서 반환하는 값은 모두 다를 수 있는 것과 같은 상황이다. 이를 Overriding이라 한다.

1. Call by value/Call by address/Call by Reference/Return by Reference

Call by value는 함수에서 값을 호출할 때 값을 복사해서 호출하는 방법이다. Call by address는 함수에서 호출을 진행할 때 변수의 주소를 복사하는 방법으로 호출하는 것이다. Call by Reference는 함수에서 호출을 진행할 때 변수의 메모리에 직접 접근하는 방법으로 호출한다. 값의 복사라는 관점에서, call by value와 call by address는 큰 차이가 없다고 봐도 무방하다. C언어에서는 사실상 call by reference와 call by address가 큰 차이 없이 쓰였지만 C++에 오면서 참조라는 새로운 문법이 생기면서 차이가 생기게 되었다.

Return by reference는 함수의 return 구문 뒤에 붙는 객체 자체를 반환하는 것이다. 일반적인 경우, return을 진행하면 그 값을 복사한 임시변수를 반환한다. 그러나 reference를 반환하면 변수인 경우 변수의 값이 아닌 자체를 변수 반환하게 된다는 차이가 있다.

1. **프로그램**
   1. 문제 선정 배경(이유)

평소에 리듬게임을 즐겨 하는데, 리듬게임은 종류가 많고 게임마다 점수 체계와 랭크 시스템이 모두 달라서 점수만을 보고는 바로바로 랭크 판별이 힘들었다. 이 때문에 점수와 게임 종류를 알면 바로바로 랭크를 출력해주는 간단한 출력 프로그램을 만들고 싶었다.

* 1. 문제 설명 및 분석

이 프로그램에서 다룰 리듬게임은 총 6종이며, 각 게임에는 영어 대문자 알파벳 4개로 구성된 약자 코드가 존재한다. 각각 JUBT (유비트), SDVX (사볼), POPN (팝픈), DDRV (디디알), MUSC (뮤제카), NSTL (노스텔지아) 이다.

이 프로그램은 간단한 출력 문제이다. 프로그램을 실행하면 메뉴가 나타나고, 메뉴는 크게 세 부분으로 나뉜다. 첫째 메뉴는 리듬게임 코드와 점수를 입력하면 랭크를 출력한다. 두 번째는 게임 코드를 입력 받으면 해당 게임의 랭크 시스템을 출력한다. 마지막은 각 게임에 해당하는 코드(약자)를 확인할 수 있도록 출력한다. 순서대로 1~3번을 입력한 경우 실행되며 0키로 종료한다. 다른 버튼을 입력한 경우 다시 메뉴창이 나타난다.

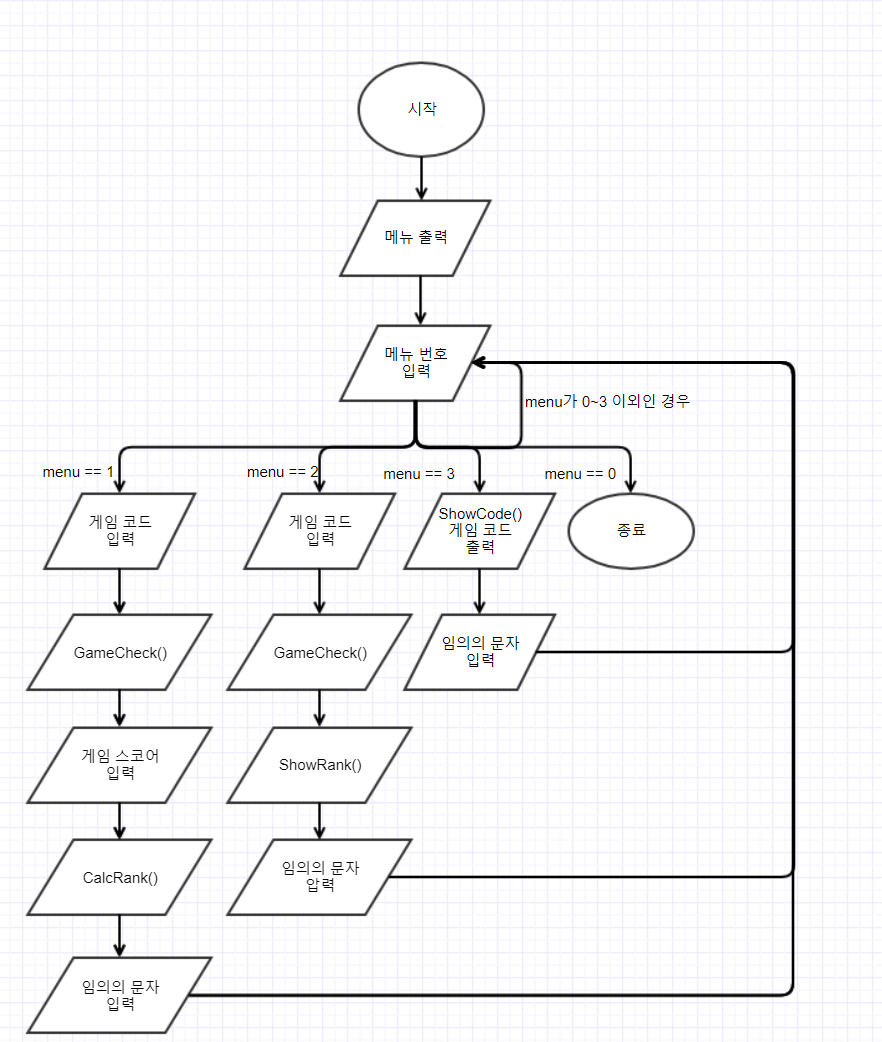
첫째 메뉴를 실행하면 먼저 게임 코드 입력창이 나타난다. 잘못된 코드를 입력하면 에러 메시지가 출력되며 다시 처음의 메뉴로 돌아간다. 올바른 코드를 입력하면 점수 입력창이 나타나고 입력 점수에 따른 랭크가 나타나게 된다. 점수 입력은 정수로만 이루어진다 가정한다.

둘째 메뉴를 실행하면 게임 코드 입력창이 나타난다. 잘못된 코드를 입력하면 에러 메시지가 출력되며 다시 처음의 메뉴로 돌아간다. 올바른 게임 코드를 입력하면, 해당 게임의 랭크 시스템을 출력한다.

셋째 메뉴를 실행하면 바로 게임에 해당하는 4글자 영문 대문자의 게임 코드 목록을 출력한다.

각 메뉴는 모두 완료되고 나면 임의의 글자를 입력(타이핑 후 엔터 입력)하면 초기 메뉴 화면으로 돌아가게 된다.

* 1. 문제 해결방법(알고리즘 도출) // 순서도, 클래스 다이어그램 등 사용 가능



<전체 프로그램의 순서도>

위와 같이 프로그램을 설계했다. GameCheck 함수는 입력받은 문자열에 대해 어떤 게임인지, 올바른 문자열인지를 판단하는 함수다. CalcRank 함수는 게임 코드와 입력받은 점수를 토대로 랭크를 출력하는 함수이다. ShowRank 함수는 입력받은 게임 코드에 대한 랭크 시스템을 출력하는 함수이다. ShowCode 함수는 게임 코드와 게임명을 나열해 출력하는 프로그램이다.

* 1. 구현 코드 설명

<main 함수>

main 함수는 전체적인 골격 역할을 한다. 먼저 변수들을 선언한다. 이후 메뉴 루프를 구현했다. 먼저 ShowMenu 함수를 이용해 메뉴를 출력한 뒤, string형 변수 menu를 입력 받는다.

1번인 경우, 간단한 메시지를 출력하고 문자열 game을 입력받는다. 이를 GameCheck 함수를 이용해 정수형 변수 gameCode로 변환한 뒤, 잘못된 입력이면 에러 메시지 출력 후 처음 메뉴로 돌아간다. 올바른 입력인 경우, 점수형 변수 score를 입력받고 입력받은 게임 코드와 점수를 토대로 CalcRank 함수를 이용해 랭크를 출력한다.

2번인 경우, 간단한 메시지를 출력하고 문자열game를 입력받는다. 입력받은 문자열을 GameCheck 함수를 이용해 정수형 변수 gameCode로 변환하고 이 값을 ShowRank 함수에 호출해 해당 게임 코드에 대응하는 게임의 랭크 시스템을 출력한다.

3번인 경우, 간단한 메시지를 출력한 후 ShowCode 함수를 이용해 전체 게임의 코드와 대응 게임을 출력한다.

0번인 경우, 그대로 프로그램을 종료한다.

그 외의 문자가 입력된 경우, 그대로 루프의 처음으로 돌아가서 다시 메뉴를 출력하고 메뉴를 입력받게 된다.

<ShowMenu 함수>

입력값이나 반환값이 없다. 함수가 실행되면 메뉴 내용을 출력하는 함수이다.

<CalcRank 함수>

정수형 변수 gameCode와 score를 입력받는 함수이고 반환값은 없다. 두 입력값을 토대로 입력한 점수가 입력한 게임에서 어떤 랭크에 해당하는지 판단해 출력한다. 입력된 점수가 올바르지 않을 경우, 계산 작업 없이 처음 메뉴로 돌아간다.

<ShowRank 함수>

정수형 변수 gameCode를 입력받는다. 반환값은 없다. 입력받은 게임에 해당하는 점수별 랭크 시스템을 출력하는 함수이다.

<ShowCode 함수>

입력값, 반환값이 없다. 영어 대문자 4자리 게임 코드와 그 코드에 대응하는 게임의 목록을 출력하는 함수이다.

<GameCheck 함수>

문자열 game을 입력받고, 정수값을 반환하게 된다. 입력받은 문자열이 우리가 사용하는 6종의 게임에 해당하지 않을 경우, 0을 반환한다. JUBT인 경우 1을, SDVX인 경우 2를, POPN인 경우 3을, DDRV인 경우 4를, MUSC인 경우 5를, NSTL인 경우 6을 반환한다.

* 1. 토론 및 결론

원래는 점수를 입력하면 랭크를 출력하는 단순한 코드 뿐만 아니라, 입력한 점수를 파일에 저장하게 하여 여러가지로 활용할 수 있게 하려고 했었다. 그리고 랭크와 게임 종류 외에도 곡명, 점수 등 다른 세부적인 데이터도 더 많이 활용하고 싶었다. 원하는 대로 되었다면, 파일에 저장된 데이터를 토대로 게임 데이터의 정렬, 갱신 여부, 레벨 별 평균 점수와 같은 다양한 기능이 담긴 프로그램이 만들어졌어야 했지만 기간 내에 구현하기 어려울 것이라 생각해 단순한 코드에 그쳐야 했던 점이 아쉬웠다. 기회가 된다면 상위 클래스를 지정해 게임별로 상속받아 구현해보고 싶다.