

# 과제 #01

데이터사이언스를 위한 컴퓨팅 1 (2022년도 1학기, M3239.005500)

Due: 2022년 3월 23일(수) 23시 59분

## 1 Compilation Process

`sqrt.c`는 실수 하나를 입력받아 제곱근을 출력하는 간단한 C 코드이다. 다음 질문들에 답하라.

### 1.1 Preprocessing

- (a) (10점) `sqrt.c`는 두 개의 헤더 파일(`stdio.h`와 `math.h`)을 `include`한다. 하지만 우리는 이 헤더 파일들을 작성한 적이 없다. 이 두 파일은 어디에 있는가? 파일을 찾아서 열어보고, 정확한 파일 경로와 파일의 줄 수를 답하라.  
(Hint: `cpp -v /dev/null`로 C preprocessor인 `cpp`의 설정들을 확인할 수 있다.)
- (b) (10점) Preprocess까지만 진행하는 `gcc` 옵션을 찾아서 `sqrt.c`를 preprocess하라. Preprocess 된 결과에서 `scanf`, `printf`, `sqrt`를 각각 찾아서 해당 부분을 보고서에 붙여넣어라.  
(Hint1: `man gcc`로 `gcc`의 옵션들을 확인할 수 있다. `preprocessing`으로 검색하면 쉽게 찾을 수 있다.)  
(Hint2: `gcc`는 preprocessing 결과를 `stdout`, 즉 화면에 직접 출력한다. 이를 파일에 저장하고 싶다면 `-o` 옵션이나 리다이렉션(`>`)을 사용하자.)
- (c) (10점) Preprocess 된 결과에 `scanf`, `printf`, `sqrt`의 실제 구현이 들어있는가? 다시 말해, `sqrt` 코드에 실제로 제곱근을 구하는 코드가 포함되어 있는가? 포함되어 있다면 코드를 간략하게 설명하고, 포함되어 있지 않다면 포함되어 있지 않은 이유를 답하라.

### 1.2 Compilation

- (a) (10점) Object file을 출력하는 `gcc` 옵션을 찾아서 `sqrt.o`를 생성하라. 생성하는데 사용한 명령을 답하라.

## 2 C++ Programming Practice

### 2.1 Concepts in OOP (60점, 부분 점수 없음)

이번 과제의 목적은 아래 그림 1과 같이 도형의 클래스들을 정의하고 각 도형을 입력 받은 위치에 그리는 프로그램을 작성하는 것이다. 도형의 인스턴스를 생성하고 도형을 그리도록 함수를 호출 하는 소스 파일인 `main.cpp`가 제공된다.

도형을 정의하는 `shapes.cpp`와 `shapes.h`를 아래와 같은 조건을 만족하도록 작성한다.

- 그림 1에서 아래쪽에 정의된 클래스는 위쪽에 정의된 클래스를 상속 받는 클래스이다. 다시 말해, `Shape`가 최상위 클래스이며 그 밑의 `Point`, `Line`, `Triangle`, `Quadrilateral`은 `Shape`를 상속 받는 클래스이다. 마찬가지로, `Equilateral`, `Isosceles`, `Scalene`은 `Triangle`을 상속 받는다.
- `Shape` 클래스는 integer type member variable인 `x_pos`와 `y_pos`를 가진다.

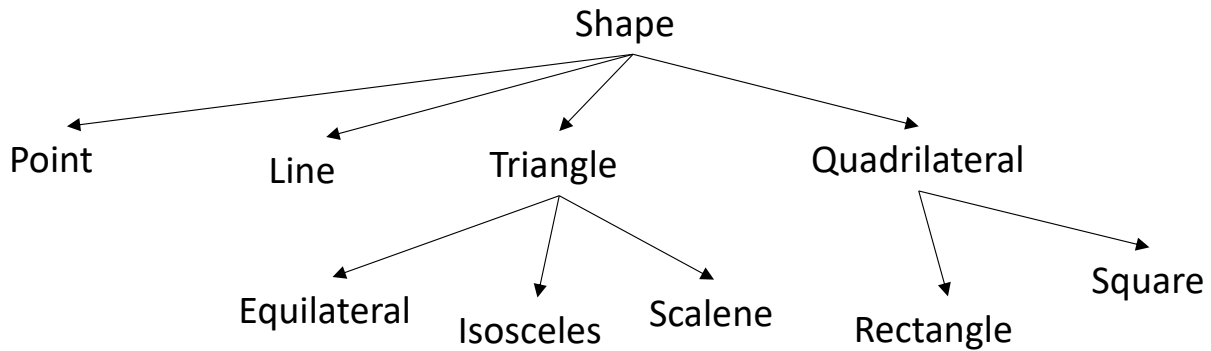


Figure 1: 도형 클래스들간의 관계

- Shape 클래스는 constructor 인자로 integer type 값 `x`와 `y`를 입력 받아 member variable인 `x_pos`와 `y_pos`에 대입한다.
- Shape 클래스는 `DrawSelf` 라는 virtual member function을 가지고 있으며, Shape 클래스를 상속 받는 모든 클래스들은 해당 function을 구현한다.
- `DrawSelf` function은 아무 인자를 받지 않으며 return type은 `void` 이다.
- `DrawSelf` function을 호출 시 "Draw {도형 종류} at ({member variable `x_pos`}, {member variable `y_pos`})" 메시지를 화면에 출력한다. 출력 예시는 아래의 결과 화면을 참고한다.

소스 코드 컴파일과 작업 스케줄러를 통한 프로그램 실행을 도와주도록 `Makefile`이 제공된다. 실습 서버에서 `make`를 입력하여 컴파일을 하고, `make test`를 입력하여 프로그램 실행 결과를 확인할 수 있다. 프로그램 실행 결과가 아래와 같으면 된다.

```

$ cd hw1
$ make clean
rm -rf oop main.o shapes.o
$ make
g++ -c -o main.o main.cpp
g++ -c -o shapes.o shapes.cpp
g++ -o oop main.o shapes.o

$ make test
salloc --nodes=1 --ntasks-per-node=1 --time=5 --cpus-per-task=1 --mem=1G ./oop
salloc: Granted job allocation 121170
Draw Square at (2, -4)
Draw Rectangle at (17, 3)
Draw Isosceles at (4, 4)
Draw Line at (-4, 10)
Draw Triangle at (8, -30)
Draw Point at (0, 0)
salloc: Relinquishing job allocation 121170
  
```

### 3 Submission Instruction

- `report.pdf`, `shapes.cpp`, `shapes.h`를 한 파일로 압축하여 ETL에 제출한다.

- `main.cpp`, `Makefile`은 수정할 수 없으며, 제출하지 않도록 한다.
- 첨부 파일명은 계정이름 `HW01.zip`으로 한다. (*e.g.*, `cfds999.HW01.zip`)
- 본인의 계정을 모를 경우 ETL 자료실에 업로드 되어있는 게시글에서 확인한다.
- 채점은 프로그램에 의해 기계적으로 처리되므로 위 사항을 지키지 않은 경우 누락되거나 불이익을 받을 수 있다.
- Grace day를 사용하려면 본인이 과제를 제출한 날에 조교에게 메일(`cfdsta@aces.snu.ac.kr`)로 알려야 한다. 메일 없이 제출만 한 경우 다음 과제를 위해 아낀 것으로 판단, 미제출 처리된다. 또한, grace day 사용 시에도 과제 제출은 이메일이 아닌 ETL을 통해 해야한다.