

# 2023 Spring OOP Assignment Report

과제 번호 : prob2

학번 : 20190445

이름 : 허수범

Povis ID : sbh408

## 명예서약 (Honor Code)

나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

I completed this programming task without the improper help of others.

프로그램을 하다 보면 결정해야 할 세부 사항이 많은데, 이러한 세부 사항을 처리한 방법과 이유를 보고서에 쓰십시오.

독창적인 아이디어와 추가 기능은 보너스 점수를 받을 수 있으므로, 보고서에 명확히 기재하십시오.

문제가 여러 개인 경우, 각 문제별로 정리해서 작성합니다.

아래 문항별 설명은 편의를 위한 것으로, 삭제하고 제출한다.

## 문제 2번

### 1. 프로그램 개요

본 프로그램은 하나의 문자(열)와 하나 혹은 두개의 실수를 입력받아 문자(열)이 지시하는 연산(사칙연산과 제곱, 제곱근)을 수행하여 결과를 출력하는 프로그램이다.

프로그램을 실행하면 입력창이 뜨고, 여기에 먼저 '+', '-', '\*', '/' 중 하나의 문자열을 입력한다면 뒤에 공백을 사이에 두고 두 개의 실수를, "sqrt" 혹은 "square"를 입력하면 뒤에 공백을 두고 하나의 실수를 입력할 수 있게 되어있고, 입력 후 해당 연산의 결과가 출력되게 되어있다.

masOS를 이용하였기 때문에 프로그램 디렉토리에는 main.cpp와 Makefile이 있다. main.cpp는 본 프로그램의 소스코드이다. Makefile은 Makefile에 작성한 명령어들을 이용하여 main.cpp를 컴파일, 테스트 컴파일, 실행시키거나 디렉토리를 정리할 수 있게 하였다. make clean으로 실행파일을 지울 수 있게 하였고, make testComp는 gcc의 컴파일 옵션을 조정하여 warning을 더 강한 조건으로 띄울 수 있게 하여 디버깅하는데에 도움을 줄 수 있도록 하였다. make all은 main.cpp가 있다면 이를 gcc로 컴파일하여 실행파일을 만들게 하고, make run은 실행파일이 있다면 프로그램을 실행하도록 하였다.

## 2. 프로그램의 구조 및 알고리즘

본 프로그램은 main함수 내에서 모두 구현되었다.

함수의 동작은 다음과 같다. 먼저 연산자를 입력할 길이 6짜리 문자열(요구되는 문자열의 최대 길이가 6이기 때문에) operat을 선언하고 초기화해주고, 두 실수 피연산자가 저장될 float형 변수 firNum과 secNum, 그리고 결과를 저장할 float형 변수 result를 선언해 주었다. 이후 각각 operat, firNum, secNum을 cin을 이용해 입력받았다. 그리고 string op를 선언하여 operat 문자 배열을 string으로 바꾸어 주었다.

이후 if else문을 이용해 연산자가 사칙연산을 가리킬 때와 제곱과 제곱근을 가리킬 때를 구분해 주었다. 연산자가 사칙연산을 가리킬 때는 cin을 이용해 secNum을 입력받도록 하였고, 또 다시 if else문을 이용해 입력이 정상적으로 이루어졌는지 확인하고(이때 cin.peek()를 활용하였다), 문제가 없는 경우 if else문을 이용해 각 사칙연산을 수행(/의 경우에는 secNum이 0일 때의 예외처리를 해주었다)하도록 하였다. 사칙연산의 결과는 result에 저장되어 소수점 세자리까지 출력하도록 하였다. 입력이 정상적이지 않은 경우 에러문을 출력하도록 하였다. 연산자가 "sqrt" 혹은 "square"일 때 역시 if else문을 이용해 정상적인 입력인지 판단하고(이 때 실수를 연산자 이후 단 하나 더 받도록 하였다. 더 입력하면 에러문이 출력되며, 숫자 하나만을 더 입력하고 엔터를 누르면 바로 결과가 출력되도록 cin.peek()를 이용해 구현하였다.) 정상인 경우 각 연산을 수행하여 출력하고(이 때 음수에 제곱근을 취하려고 하면 결과 출력에 i를 더해주도록 하였다), 정상적이지 않은 경우 에러문을 출력하도록 하였다. 연산자가 사칙연산과 제곱근 모두 가리키지 않을 경우에도 에러문을 출력하도록 하였다.

## 3. 토론 및 개선

문제2 역시 문제1과 마찬가지로 예외처리에 집중했다. 특히 제곱근과 제곱 연산자가 입력될 때 EOF를 어떻게 구현하는지가 문제였는데, 이를 cin.peek()을 이용해 구현해 보았다. 가장 간단할 것이라 생각했던 문제2에서 예외처리에 시간이 가장 오래 걸렸다. Cpp의 입력법과 syntax가 익숙하지 않은 상태로 프로그램을 작성하다보니 자잘한 실수가 많았는데, 문제2부터 gcc의 옵션들(-W, -Wall)을 이용해 warning 메시지들을 하나하나 읽어보고, gdb를 이용하며 디버깅을 해보며 해결해 나갔다. 다양한 디버깅 툴들을 이용해 보며 문제를 해결하는 것이 무작정 코드를 작성하고 실행해보는 것보다 효과적이라는 것을 알 수 있었다.

본 프로그램에 추가할 수 있는 기능은 더 많은 실수와 연산자를 이용해 다양한 연산을 수행할 수 있게 하여 실제로 쓰는 계산기와 같이 긴 줄을 입력받아 결과를 출력하거나 연속적으로 연산이 수행 가능하도록 할 수 있을 것 같다. 문자열로 전체를 받고 공백을

감지하여 각 부분으로 나눈 후 계산하거나 loop를 이용해 exit 명령을 입력할 때까지 계속 수를 더하거나 곱할 수 있도록 만들 수 있을 것이다. Class를 이용하면 긴 줄을 입력받는 경우 괄호나 각 연산자 간의 우선순위를 구현해 볼 수도 있을 것이다.

#### 4. 참고 문헌

<https://modoocode.com/194>

<https://stackoverflow.com/questions/41020417/cin-peek-for-input-validation>

(cin.peek()에 대한 자료)

[https://velog.io/@nada\\_dbstkddl/C-%EC%86%8C%EC%88%98%EC%9E%90%EB%A6%AC%EC%88%98-%EB%8B%A4%EB%A3%A8%EA%B8%B0-precision-fixed](https://velog.io/@nada_dbstkddl/C-%EC%86%8C%EC%88%98%EC%9E%90%EB%A6%AC%EC%88%98-%EB%8B%A4%EB%A3%A8%EA%B8%B0-precision-fixed) 소수점 출력