­

**Infinite Int Calculator Report**

**CAU 23-2**

**OOP Project02**

|  |  |
| --- | --- |
|  |  |
| Subject | Object Oriented Programming |
| Professor | Bong-Soo Sohn |
| Major | Computer Science and Engineering |
| Team | TEAM 07 |
| Members | 20204946 이규성  20203436 이병구  20203458 조영호 |

Contents

[(a) Summary - 3 -](#_Toc150091577)

[(b) Build and Run - 3 -](#_Toc150091578)

[Build - 3 -](#_Toc150091579)

[Run - 3 -](#_Toc150091580)

[Test environment - 3 -](#_Toc150091581)

[(c) Important Functionality - 4 -](#_Toc150091582)

[(d) how you implemented (important implementation issues) - 4 -](#_Toc150091583)

[(e) the result of SW system design - 4 -](#_Toc150091584)

[(f) execution results - 4 -](#_Toc150091585)

[(g) Appling object oriented concepts and Learning things - 4 -](#_Toc150091586)

[(h) Conclusion - 4 -](#_Toc150091587)

## Summary

Presentation speaker: 이규성

Project github Link : <https://github.com/Tastypotato245/OOP_Project02>

Description: This program can infinite integer calculation.

Feature:

* inf\_int can be made from normal int, string, and another inf\_int instance.
* Assignment overloading, copy constructor, destructor
* Comparison operation between inf\_int (==, !=, >, <)
* Three basic operation between inf\_int (+, -, \*)
* Calculation absolue value (abs())
* inf\_int can be printed (<<)

## (b) Build and Run

### Build

$ make

### Run

$ ./inf\_int 9999999999999999999 + 999999

$ ./inf\_int 12345678901234567890 \\* 123456789

$ ./inf\_int

Input: 54 \* 123456788888

Output: 6666666599952

Input: 2345234523525345 + 22233322

Output: 2345234545758667

Input: ^C (terminating)

Input: 0 (terminating)

### Test environment

* macOS 14.0 arm64 (clang 15.0.0)
* Debian GNU/Linux 11 aarch64 (g++ 10.2.1)

## (c) Important Functionality

Through the method of storing numbers in reverse order using a dynamically assigned array of characters, integers are stored and processed without limitation to the size of the numbers.

The algorithm was implanted to efficiently manage carry and borrow that can occer during calculation.

Positive and negative numbers are accurately classified through sign processing, and the sign of the operation result is determined. Can also be calculated for many numbers and signs.

## (d) how you implemented (important implementation issues)

## (e) the result of SW system design

UML

## (f) execution results

예시 사진과 결과

## (g) Appling object oriented concepts and Learning things

객체 지향 개념의 적용:

생성자와 소멸자, 연산자 오버로딩을 통해 캡슐화와 추상화를 구현하였다.

복사 생성자와 대입 연산자를 통해 객체의 올바른 복사를 보장한다.

클래스 멤버 함수를 통해 데이터에 대한 접근과 조작의 제어를 향상시킨다.

프로젝트를 통해 느낀 점 및 배운 점:

## (h) Conclusion