Project 2 Report

Garrett Nix

2-03-2026

1 Introduction

Due to memory contraints, and a need for fast implementation, object-oriented programming provides an effective method for acheiving this. This efficacy can be shown in a multitude of ways, but a simple example can be shown using a fraction.

2 Purpose

A fraction class is a useful example of object-oriented programming as it provides a resuable set of code, equipped with the methods and properties inherent to said mathematical object.

3 Implementation

The implementation I went with was what I consider a more mathematically defined approach. I used the laws afforded to rational numbers, for instance when multiplying fractions $a/b, c/d$, this results in ac/bd. Because of this, the Fraction class itself has private variables denoting the initial numerator and denominator, denoted a, b. Each method then makes use of another fraction object as an argument, and this acts as the fraction c/d. The reason for this approach was to make the use of this class more simplistic from a user perspective. The user simply constructs a fraction and then can state they want to multiply fraction A by fraction B without worry for what the numerator or denominator is for each after the initial construction.

4 Discussion

Some potential errors may arise from the fact that much of the program is not accessible by the user, having just the fraction methods specified. This could cause some friction in terms of the users ability to customize certain aspects of the fraction class in later applications.