For this assignment, you will design and implement a rational class, for storing and manipulating rational numbers of the form numer / denom, where numer and denom are long integers, and denom is not zero. The class should have the following features.

1. **(15 points)** There is a constructor or multiple constructors so that a rational may be constructed in the following ways:
   * with no parameters (initialized to zero),
   * with a single long integer parameter (initialized to that integer),
   * with long integer parameters for the numerator and denominator (you may assume the denominator is not 0).

The constructor(s) is/are responsible for simplifying the fraction so that the GCD of numer and denom is 1, and denom is positive. For example, rational(6, -3) should reduce to -2/1. Note that the fraction may be improper.

1. **(20 points)** Operators +, -, \*, / are overloaded.
2. **(5 points)** The stream output operator is overloaded so that a rational object may be placed onto an ostream object (such as cout). If the denominator is 1, then the rational should be displayed as an integer; otherwise, the rational should be displayed as a fraction. If the fraction is improper, it should be displayed as such. For example, display 5/3, not 1 2/3.
3. You may write private helper methods as needed to implement the above features.
4. **(5 points)**Your class definition should be submitted in a header file, rational.hh while all methods should be implemented in rational.cc.

To test your implementation, you may use calc.cc (see the git repository for class, under Homeworks) which includes rational.hh and implements a simple postfix calculator. Running

./calc 4 3 / 7 6 / +

should display 5/2. Submit your source files, rational.hh and rational.cc in Canvas.