Recitation Worksheet: Week 3 – Design / Operator Overload

*This recitation will be graded based on group participation. Write your name on the attendance sheet.

1. Form small groups of 3-4 with students near you. During approximately the first 20 minutes your group will design a simple class to represent rational numbers with your group. If you need a refresher on what defines a rational number, check out https://en.wikipedia.org/wiki/Rational_number.

   a. Design, on paper, a rational class. Start with a high-level design for the class. What data members will it have? What member functions will it have and what will the return types and parameter types for these member functions be? Now, write complete pseudo-code for the following member functions:

      i. A default constructor that sets the numerator and denominator to 1.
      ii. A non-default constructor with two parameters for numerator and denominator
      iii. Getters/Setters for all data members.
      iv. Operator Overloaded functions for * and / operators.
      v. A member function that reduces the rational number to its most reduced form.
      vi. If you finish the above before other groups, how might you overload the << operator to allow the

   b. Take the next 5-10 minutes to design a plan for how you would write a driver (main) to test your rational class functionality. What are the different cases you need to consider to be sure your class is robust? Adjust your design if needed. List out the tests you will perform, and what the expected outcome of these tests should be.

2. How might you think about setting up the design using inheritance Rational, Integer, and Natural number sets (the Q, Z, N shown below)?

   R Q Z N

   Talk about this in your group. What is the base class? What data do all the number sets have in common that can be included in the base class? What changes with the child classes and how would you design for this?