CSSE 221: Fundamentals of Software Development Honors

Programming assignment (due by Saturday 11:59 pm)

Finish BigRational, as stated in the specification, being sure you have checked in your final, fully-documented copy to the repository.

Written homework to do before week 2 (due before beginning of class on Monday, Sept 10).

As usual, please bring hardcopy for written answers to class to hand in on Monday.

1. Be sure you've done all reading for this week.
2. I want to compare two BigRationals, x and y, to see if they are equal value (intuitively, x == y).
   a. Write an expression for this using the compareTo() method.

      if (x.compareTo(y) = = 0)

   b. Write an expression for this using the equals() method.

      if (x.equals(y))

   c. What is compared if instead I write (x == y)? (Hint, you saw this in section 2.2. This is important to understand!)

      This compares memory addresses, not values in the objects. Checking whether or not it's the same object, not if it has the same value.

3. For the sake of simplicity, on this question, you may ignore effects such as operating system interrupts that may change the runtime slightly. (Focus on the meaning of big-oh only.)
   a. A block of code is characterized as O(n) and runs in 23 milliseconds. If n is increased by a factor of 8, what is the new runtime of the code?

      184 milliseconds

   b. A block of code is characterized as O(n^2) and runs in 400 milliseconds. If n is increased by a factor of 3, what is the new runtime of the code?

      3,600 milliseconds

4. What combination of control structure(s) gives code that is O(n^3)?

Three nested loops

5. How many times is the sum statement executed (in terms of *n*):

```
for (int i = 0; i < n; i++) {

  for (int j = 0; j < n; j++) {
         sum++;
        }
        for (int j = 0; j < n; j++) {
          sum++;
        }
}
```

Exact answer = _**$2n^2$**__, giving an asymptotic answer of O(_**n^2**__).

6. Name at least 3 advantages of unit testing.

Will vary

7. Suppose a user of your program finds a bug. What action should you take beyond fixing the bug?

Answer provided, no credit

8. Write the code for a method that does the following. Note, you may choose to type this into Eclipse to test it, but you aren't required to do so:

   Pass into the method a BigRational object. Evaluate the object to determine if it is equal to its own absolute value. If it is, print the message "Good going", and terminate the program. If it is not, print the message "Not good going", and terminate the program. Regardless of the outcome of this test, print the message "Done" before terminating.

   Write this code twice: once using if-then-else block(s) to evaluate the data and handle a possible error, and once using a try-catch block to evaluate the data and handle a possible error.

   Note: This code would never be written using a try-catch block in a production environment; we do it here to make sure you understand important structures.

```
public void equalToAbsoluteValueIf(BigRational br) {
      if (br!=null) {
            if (br.equals(br.abs())) {
                  System.out.println("Good going");
            }
            else {
                  System.out.println("Not good going");
            }
      }
      System.out.println("Done");
}
```

```
public void equalToAbsoluteValueTry(BigRational br) {
      try {
            if (br.equals(br.abs())) {
                  throw new Exception("Good going");
            }
            else {
                  throw new Exception("Not good going");
            }
      }
      catch (NullPointerException e) {}
      catch (Exception e) {
            System.out.println(e.getMessage());
      }
      finally {
            System.out.println("Done");
      }
}
```