

CSSE 221: Fundamentals of Software Development Honors

Written homework

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.

5 POINTS POSSIBLE	
Answer	Point Value
if (x.compareTo(y) == 0) OR if (y.compareTo(x) == 0)	Full Credit: 5/5
x.compareTo(y) == 0 OR y.compareTo(x) == 0	Full Credit 5/5
Missing parenthesis around if condition: if x.compareTo(y) == 0 OR if y.compareTo(x) == 0	-2
Single equal sign	-2
Incorrect number after ==	-1
.compareTo without parenthesis	-2
Misspelled compareTo	-1
Anything else	0/5 (or best judgment)

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

5 POINTS POSSIBLE	
Answer	Point Value

if (x.equals(y)) OR if (y.equals(x))	Full Credit: 5/5
x.compareTo(y) OR y.compareTo(x)	Full Credit 5/5
Missing parenthesis around if condition: if x.equals(y) OR if y.equals(x) == 0	-2
.equals without parenthesis	-2
Misspelled equals	-1
Anything else	0/5 (or best judgment)

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

10 POINTS POSSIBLE
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?

10 POINTS POSSIBLE
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?

10 POINTS POSSIBLE
3,600 milliseconds

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

10 POINTS POSSIBLE
Triply-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 $O(n^2)$.

5 POINTS POSSIBLE	
Answer	Points Value
$2n^2$	5/5
Work shown (and approaching correct answer), answer is incorrect	2/5

5 POINTS POSSIBLE	
Answer	Points Value
$O(n^2)$	5/5
Work shown (and approaching correct answer), answer is incorrect	2/5

6. 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");
}

```

20 POINTS POSSIBLE	
Answer	Point Value
“Done” println is in both if and else OR “Done” is not printed at all	-5
“Done” only prints in certain cases	-3
Using this instead of parameter br	-5
Forgetting to print “Done”	-5
Prints the wrong message at the wrong time	-5
Messages are misspelled/incorrect	-3
Zero Functionality	-20

```

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");
    }
}

```

20 POINTS POSSIBLE	
Answer	Point Value
“Done” print is not within the finally but only prints once each time	-2
“Done” only prints in certain cases	-3
Using this instead of parameter br	-5

Forgetting to print "Done"	-5
Prints the wrong message at the wrong time	-5
Messages are misspelled/incorrect	-3
Printing is done in if/else not in catch/finally	-10
Zero Functionality	-20