

**NAME:** Abenes, Enrico O.  
**COURSE:** CC1 INTL

**DATE:** May 23, 2023  
**SCHEDULE:** 1: 30 pm – 5: 20 pm MT

**TITLE:**        **Relational Operators**

**LEARNING OBJECTIVES:**

At the end of this activity, the students should be able to:

1. Identify basic relational operations available in Java language.
2. Perform the basic relational operations as implemented in a Java program.
3. Determine the precedence or order of evaluation of these operators.
4. Create a complete Java program that simulates these basic operations.

**INSTRUCTIONS:**

1. Make sure you have your own individual account.
2. Always keep your account secret to others to avoid unauthorized access to your files.
3. Always save your work and log-off when not using the computer.
4. By now you should have been familiarized using your text editor.
5. By now you should know how to create, save, compile, execute, and debug programs in Java.
6. Use the skills and learning obtained in Prelim Laboratory Activity 1 to Prelim Laboratory Activity 3 in order for you to successfully finish the learning objectives of this module.

**DURATION: One to two Meetings**

**HANDS-ON:**

1. Log-on using your own individual account. Use your own **username** and **password**.
2. Open your text editor.
3. Write your next Java program:
  - 3.1. Write your next program by copying the source code shown below to your text editor.

```
/* Programmed by: <write your name here>
   Program title: RelationalOperators.java
   Program Date: <write the date today here> */

public class RelationalOperators{
    public static void main(String[] args){

        boolean a, b, c, d, e, f;
        int x, y;
```

```

        x = 8;
        y = 13;

        a = (x == y);
        b = (x != y);
        c = (x > y);
        d = (x < y);
        e = (x >= y);
        f = (x <= y);

        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("c = " + c);
        System.out.println("d = " + d);
        System.out.println("e = " + e);
        System.out.println("f = " + f);

    }
}

```

- 3.2. Save your program as **RelationalOperators.java** then compile your program until no errors and warnings are reported.
- 3.3. Execute your program.
- 3.4. Write what will be displayed on the screen.

```

/* Programmed by: Abenes, Enrico O.
   Program Title: RelationalOperators.java
   Program Date: May 23, 2023 */

public class RelationalOperators {
    public static void main(String[] args) {

        boolean a, b, c, d, e, f;
        int x, y;

        x = 8;
        y = 13;

        a = (x == y);
        b = (x != y);
        c = (x > y);
        d = (x < y);
        e = (x >= y);
        f = (x <= y);

        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("c = " + c);
        System.out.println("d = " + d);
        System.out.println("e = " + e);
        System.out.println("f = " + f);

    }
}

```

```

a = false
b = true
c = false
d = true
e = false
f = true

```

- 3.5. Assuming that the values for  $x$  and  $y$  were changed, interpret the value that would be generated whether it is TRUE or FALSE

```

x = 27;
y = 15;

```

<code>a = (x == y);</code>	$\rightarrow$	True or False?	<u>FALSE</u>
<code>b = (x != y);</code>	$\rightarrow$	True or False?	<u>TRUE</u>
<code>c = (x &gt; y);</code>	$\rightarrow$	True or False?	<u>TRUE</u>
<code>d = (x &lt; y);</code>	$\rightarrow$	True or False?	<u>FALSE</u>
<code>e = (x &gt;= y);</code>	$\rightarrow$	True or False?	<u>TRUE</u>
<code>f = (x &lt;= y);</code>	$\rightarrow$	True or False?	<u>FALSE</u>

4. Simulate the following code segment in Java and answer the following questions based on the relational operators being performed. Write as your answer whether the statement is TRUE or FALSE. Each number is independent of the other.

	True / False
1. ( <code>'A' &lt; 'B'</code> )	<u>TRUE</u>
2. ( <code>713 == 317</code> )	<u>FALSE</u>
3. ( <code>3+2+1 &gt;= 1+2+3</code> )	<u>TRUE</u>
4. ( <code>3.0 &lt;= 6.0/2.0</code> )	<u>TRUE</u>
5. ( <code>2+3*4 != 2*3+10</code> )	<u>TRUE</u>
6. ( <code>'Z' == 90</code> )	<u>TRUE</u>
7. ( <code>'Y' == 'y'</code> )	<u>FALSE</u>
8. ( <code>10%2 &gt; 5%3</code> )	<u>FALSE</u>

5. Modify the `main()` block of the program by changing the statements with the one written below.

```

x = 17;
y = 20;

a = ((x - y) > (y - x));
b = ((++x) == (x+1));
c = ((y%x) <= (y-x));
d = ((x%y) > (y%x));
e = ((x * 2) != (2 * x));
f = ((x-1) > (--x));
System.out.println("a= " + a);
System.out.println("b= " + b);
System.out.println("c= " + c);
System.out.println("d= " + d);
System.out.println("e= " + e);
System.out.println("f= " + f);

```

5.1. Save, Compile and Execute and write the output on the box provided below.

```

a= false
b= false
c= true
d= true
e= false
f= false

```

5.2. List all relational operators that were used in the program and describe the function of each briefly.

1. **Greater than (>):** If the value on the left is greater than the value on the right, the comparison is made. If the condition is met, returns true; otherwise, returns false.
2. **Equal to (==):** This function determines whether the value on the left is equal to the value on the right. If the condition is met, returns true; otherwise, returns false.
3. **Less than or Equal to (<=):** This function determines whether the value on the left is less than or equal to the value on the right. If the condition is met, returns true; otherwise, returns false.
4. **Not Equal to (!=):** If the value on the left is not equal to the value on the right, the comparison fails. If the condition is met, returns true; otherwise, returns false.

5.3. Given the following expression,

```

statement = (((a + b) * c) >= ((d - e) % (a + b
* c))) == ((++d) < (--e))

```

identify ALL operators used in the expression and determine the precedence on how these operators are going to be performed.

1. **Parenthesis ()**: Expressions included in parenthesis are evaluated first.
2. **Increment & Decrement Operators ++, --**: Following that, the pre-increment (++d) and pre-decrement (--e) operators are assessed.
3. **Arithmetic Operators \*, /, %**: Following that, the multiplication (b \* c) and modulus ((d - e) % (a + b \* c)) operations are evaluated.
4. **Arithmetic Operators +, -**: The following operations are evaluated: addition (a + b) and subtraction ((d - e) - (a + b \* c)).
5. **Relational Operators >=, <**: The comparisons higher than or equal to ((a + b) \* c) >= ((d - e) % (a + b \* c)) and less than ((++d) (--e)) are evaluated next.
6. **Relational Operators ==**: Last, the equality comparison (((a + b) \* c) >= ((d - e) % (a + b \* c))) == ((++d) (--e)) is performed.

