3. Suppose that `a` and `b` are `boolean` values. Show that the expression `(!(a && b) && (a || b)) || ((a && b) || !(a || b))` is equivalent to `true`.

```
boolean a = true;
boolean b = true;

boolean a = true;
boolean b = false;

boolean a = false;
boolean b = true;

boolean a = false;
boolean b = false;
```

4. Suppose that `a` and `b` are `int` values. Simplify the following expression: `(!(a < b) && !(a > b))`

Solution: `(a == b)`

6. Why does `10/3` give `3` and not `3.33333333`?

Solution. **Since both 10 and 3 are integer literals, Java sees no need for type conversion and uses integer division. You should write `10.0/3.0` if you mean the numbers to be `double` literals. If you write `10/3.0` or `10.0/3`, Java does implicit conversion to get the same result.**

7. What do each of the following print?
   - **a.** `System.out.println(2 + "bc");` **prints: 2bc**
   - **b.** `System.out.println(2 + 3 + "bc");` **prints: 5bc**
   - c. `System.out.println((2+3) + "bc");` **prints: 5bc**
   - **d.** `System.out.println("bc" + (2+3));` **prints: bc5**
   - **e.** `System.out.println("bc" + 2 + 3);` **prints: bc23**

9. What do each of the following print?
   a. `System.out.println('b');`

      `b` **print directly character b**

   b. `System.out.println('b' + 'c');`

      197 prints the sum of integers values of letter b and c (98+99=197)

   c. `System.out.println((char) ('a' + 4));`

      e int value of letter ‘a’ adding 4 (97+4=101) and prints the char cast of this value (char (101))

Explain each outcome.

10. Suppose that a variable `a` is declared as `int a = 2147483647` (or equivalently, `Integer.MAX_VALUE`). What do each of the following print?
    a. `System.out.println(a);`

       2147483647 **prints original value**

    b. `System.out.println(a + 1);`

       -2147483648 **Java uses two's complement to represent the various forms of integers and cause Integer overflows in this case**

    c. `System.out.println(2 - a);`

       -2147483645 **Java uses two's complement to represent the various forms of integers and cause Integer overflows in this case**

    d. `System.out.println(-2 - a);`

       2147483647 **Java uses two's complement to represent the various forms of integers and cause Integer overflows in this case**

    e. `System.out.println(2 * a);`

       -2 **Java uses two's complement to represent the various forms of integers and cause Integer overflows in this case**

    f. `System.out.println(4 * a);`

       -4 **Java uses two's complement to represent the various forms of integers and cause Integer overflows in this case**

Explain each outcome.

11. Suppose that a variable `a` is declared as `double a = 3.14159`. What do each of the following print?
    a. `System.out.println(a);`

       3.14159 **prints the original result**

b. `System.out.println(a + 1);`

4.14159 **prints the original result (adding 1 integer value)**

c. `System.out.println(8 / (int) a);`

2 **prints the result of 8/2 = 4**

d. `System.out.println(8 / a);`

2.5464812403910124 **prints the original result**

e. `System.out.println((int) (8 / a));`

2 **prints the result of 2.5464812403910124 on int value**

Explain each outcome.

20. Write a program SumOfTwoDice.java that prints the sum of two random integers between 1 and 6 (such as you might get when rolling dice).

29. Day of the week. Write a program DayOfWeek.java that takes a date as input and prints the day of the week that date falls on. Your program should take three command-line arguments: `m` (month), `d` (day), and `y` (year). For `m` use 1 for January, 2 for February, and so forth. For output print 0 for Sunday, 1 for Monday, 2 for Tuesday, and so forth. Use the following formulas, for the Gregorian calendar:

```
y0 = y - (14 - m) / 12
x = y0 + y0/4 - y0/100 + y0/400
m0 = m + 12 * ((14 - m) / 12) - 2
d0 = (d + x + (31*m0)/ 12) mod 7
```

For example, on what day of the week was August 2, 1953?

```
y = 1953 - 0 = 1953
x = 1953 + 1953/4 - 1953/100 + 1953/400 = 2426
m = 8 + 12*0 - 2 = 6
d = (2 + 2426 + (31*6) / 12) mod 7 = 2443 mod 7 = 0   (Sunday)
```

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

```
        int m = Integer.parseInt(args[0]);
        int d = Integer.parseInt(args[1]);
        int y = Integer.parseInt(args[2]);

        int y0 = y - (14 - m) / 12;
        int x = y0 + y0/4 - y0/100 + y0/400;
        int m0 = m + 12 * ((14 - m) / 12) - 2;
        int d0 = (d + x + (31*m0)/12) % 7;

        String[] strDays = new String[] { "Sunday", "Monday", "Tuesday", "Wednesday",
"Thusday", "Friday", "Saturday" };
        java.util.Calendar c = Calendar.getInstance();
        c.set(Calendar.MONTH, m-1);
        c.set(Calendar.DAY_OF_MONTH, d);
        c.set(Calendar.YEAR, y);

        System.out.println(d0);
        System.out.println(strDays[d0]);
        System.out.println(c.getTime());
    }
}
```