Decisions in Java – Comparing Values

In order to make decisions, Java uses the concept of `true` and `false`, which are *boolean values*.

Boolean Expressions

A *boolean expression* is similar to a mathematical expression, except that the result is `true` or `false`, rather than a numeric value. To create a boolean expression, we use the relational operators to compare the values of various data types, such as integers, floats, characters, or strings, using a *relational expression*.

Relational Operator	Meaning	Example	Result
==	is equal to	5 == 5	TRUE
!=	is not equal to	5 != 5	FALSE
<	is less than	3 < 7	TRUE
<=	is less than or equal to	4 <= 4	TRUE
>	is greater than	3 > 7	FALSE
>=	is greater than or equal to	7 >= 3	TRUE

The following rules apply to the use of relational operators with different data types:

1. Values of any of the primitive numeric data types (e.g., int, float, and all their variations) can be used with any of the relational operators.
2. Boolean data types can only be tested as “equal to” or “not equal to”.
3. Values of type `char` are ordered according to the Unicode encoding system. A character the occurs earlier in the system is “less than” a character that occurs later in the system. You can research full details of the Unicode system online.
   a) For alphabetic characters, this means that 'a' is less than 'z', and 'A' is less than 'Z', as expected.
   b) In the Unicode system, all uppercase letters occur earlier than all lowercase letters. Thus we get the relational ordering of:
   `'A' < 'B' < 'C' < ... < 'Z' < 'a' < 'b' < 'c' < ... <'z'`
   c) Representing numbers as characters, such as when you type on a keyboard, keeps the same ordering, so that `'0' < '1' < '2' < ... < '9'`.

Decisions in Java – Comparing Values

Exercises

1. For each of the legal `boolean` expressions, state its value (true or false). For each illegal expression, state the reason that it is illegal.

   (a) `-3 != 3`
   (b) `'Q' == 'q'`
   (c) `'m' < = 'p'`
   (d) `'*' < '*'`
   (e) `8.23 =< 8.2300`
   (f) `(7/3) = 2`
   (g) `false == 0`
   (h) `(25 % 4) >= 1`

2. For each expression, state whether it is true or false.

   (a) `'q' < 'm'`
   (b) `'G' > 'K'`
   (c) `'a' < 'Z'`
   (d) `'5' < 'v'`
   (e) `'q' > '7'`
   (f) `'9' < ' '`
   (g) `'X' < 'y'`
   (h) `'i' < 'I'`

3. Determine the value of each expression.

   (a) `17 / 3 < 17 / 3.0`
   (b) `'F' > 'B' + 3`
   (c) `6 % 3 > 0`
   (d) `(2 + 3 < 6 ) == true`
   (e) `(2 * 3) < 5 != true`