

## Character Component — A Brief Summary

**Range of values:** ASCII characters (see the ASCII Character Set)

**Initial value:** `\0` (this two-character “escape sequence” denotes the *null* character)

**Assignment operator:** `=`

**Relational operators:**

|                    |                              |
|--------------------|------------------------------|
| <code>==</code>    | (equal)                      |
| <code>!=</code>    | (not equal)                  |
| <code>&lt;</code>  | (less than and not equal)    |
| <code>&lt;=</code> | (less than or equal)         |
| <code>&gt;</code>  | (greater than and not equal) |
| <code>&gt;=</code> | (greater than or equal)      |

**Escape sequences:**

|                 |                             |
|-----------------|-----------------------------|
| <code>\n</code> | newline                     |
| <code>\t</code> | horizontal tab              |
| <code>\\</code> | backslash character         |
| <code>\'</code> | single quote-mark character |

**Input and output:**

Assume that `input` is an object of type `Character_IStream`, that `output` is an object of type `Character_OStream`, and that `c` is an object of type `Character`.

- To input a value for `c` use `input >> c`.
- To output the value of `c` use `output << c`.

**Conversion operators:**

Assume that `c` is an object of type `Character`.

- To convert the value of `c` to a text string use `To_Text (c)`.
- To convert the value of `c` to an integer use `To_Integer (c)`.

Note: The value of `To_Integer (c)` is the ASCII code for the character `c`.