Characters

Individual characters in C++ are represented by the built-in primitive data type named `char` (usually pronounced "tchar", not "kar"). In memory, these values are represented by assigning each character an 8-bit integer code called an ASCII code . (Actually, only 7-bits are defined by C++, so the ASCII values 128-255 are non-standard and may vary from platform to platform.)

You write **character literals** by enclosing each character **in single quotes**. Thus, the literal `'A'` represents the internal code of the uppercase letter **A**.

In addition, C++ allows you to write **special characters** in a multi-character form beginning with a back-slash (`\`). This form is called **an escape sequence**. This includes the **newline** (`\n`), the **tab** (`\t`), and a double-quote inside a string literal (`\"`). Here is a list of the C++ escape sequences .

Character Functions

It is useful to have tools for working with individual characters. The `<cctype>` header contains a variety of functions that do that. There are two kinds of functions.

- **Predicate classification functions** test whether a character belongs to a particular category. Calling `isdigit(ch)` returns true if `ch` is one of the digit characters in the range between `'0'` and `'9'`. Similarly, `isspace(ch)` returns true if `ch` is any of the characters that appear as white space on a display screen, such as spaces and tabs.
- **Conversion macros** make it easy to convert between uppercase and lowercase letters. Calling `toupper('a')`, for example, returns the character `'A'`. If the argument is **not** a letter, the function returns it unchanged, so that `tolower('7')` returns `'7'`.