newline into the source, i.e., making the string literal span lines of source code, is an error. In contrast to conventional string literals, *raw* string literals (1) treat unescaped embedded double quotes (") as literal data, (2) do not interpret special-character escape sequences (e.g., \n, \t), and (3) interpret both vertical and horizontal whitespace characters present in the source file as part of the string contents:

```
const char s1[] = R"(line one
line two
    line three)";
    // OK
```

In this example, we assume that all trailing whitespace has been stripped since even trailing whitespace in a raw literal would be captured. Note that any literal tab characters are treated the same as a \t and hence can be problematic, especially when developers have inconsistent tab settings; see *Potential Pitfalls — Unexpected indentation* on page 112. Finally, all string literals are concatenated with adjacent ones in the same way the conventional ones are in C++03:

```
const char s2[] = R"(line one)"         "\n"
                    "line two"          "\n"
                  R"(    line three)";
    // OK, equivalent to "line one\nline two\n    line three"
```

These same rules apply to both raw *wide* string literals and raw *Unicode* ones (see Section 1.1. "Unicode Literals" on page 129) that are introduced by placing their corresponding prefix before the R character:

```
const wchar_t  ws [] =  LR"(Raw\tWide\tLiteral)";
    // represents "Raw\tWide\tLiteral", not "Raw    Wide    Literal"


const char     u8s[] = u8R"(\U0001F378)";  // Represents "\U0001F378", not "🍸"
const char16_t us [] =  uR"(\U0001F378)";  //      "           "        "    "
const char32_t Us [] =  UR"(\U0001F378)";  //      "           "        "    "
```

Collisions

Although unlikely, the data to be expressed within a string literal might itself contain the character sequence)" embedded within it:

```
#include <cstdio>   // printf

void emitHelloWorld()
{
    printf("printf(\"Hello, World!\")");
  //                               ^^
  // The )" character sequence terminates a typical raw string literal.
}
```