In C++, the `for` and `while` loops essentially are equivalent. For example, the `for` loop

```cpp
for (init-expression; test-expression; update-expression)
{
    statement(s)
}
```

could be rewritten this way:

```cpp
init-expression;
while (test-expression)
{
    statement(s)
    update-expression;
}
```

Similarly, the `while` loop

```cpp
while (test-expression)
    body
```

could be rewritten this way:

```cpp
for ( ; ; test-expression;)
    body
```

The `for` loop requires three expressions (or, more technically, one statement followed by two expressions), but they can be empty expressions (or statements). Only the two semicolons are mandatory. Incidentally, a missing test expression in a `for` loop is construed as true, so the loop

```cpp
for ( ; ; )
    body
```

runs forever.

Because the `for` loop and `while` loop are nearly equivalent, the one you use is a matter of style. (There is a slight difference if the body includes a `continue` statement, which is