public member function

std::istream::tellg

<istream> <iostream>

```
streampos tellg();
```

Get position in input sequence

Returns the position of the current character in the input stream.

Internally, the function accesses the input sequence by first constructing a `sentry` object (with *noskipws* set to `true`) without evaluating it. Then, if member `fail` returns `true`, the function returns `-1`.
Otherwise, returns `rdbuf()->pubseekoff(0,cur,in)`. Finally, it destroys the `sentry` object before returning.

Notice that the function will work even if the `eofbit` flag is set before the call.

Calling this function does not alter the value returned by `gcount`.

Parameters

none

Return Value

The current position in the stream.
If either the *stream buffer* associated to the stream does not support the operation, or if it fails, the function returns `-1`.
`streampos` is an `fpos` type (it can be converted to/from integral types).

Errors are signaled by modifying the *internal state flags*:

flag	error
`eofbit`	-
`failbit`	The construction of `sentry` failed (such as when the *stream state* was not `good` before the call).
`badbit`	Error on stream (such as when this function catches an exception thrown by an internal operation). When set, the integrity of the stream may have been affected.

Multiple flags may be set by a single operation.

If the operation sets an *internal state flag* that was registered with member `exceptions`, the function throws an exception of member type `failure`.

Example

```
// read a file into memory
#include <iostream>     // std::cout
#include <fstream>      // std::ifstream

int main () {
  std::ifstream is ("test.txt", std::ifstream::binary);
  if (is) {
    // get length of file:
    is.seekg (0, is.end);
    int length = is.tellg();
    is.seekg (0, is.beg);

    // allocate memory:
    char * buffer = new char [length];

    // read data as a block:
    is.read (buffer,length);

    is.close();

    // print content:
    std::cout.write (buffer,length);

    delete[] buffer;
  }

  return 0;
}
```

In this example, `tellg` is used to get the position in the stream after it has been moved with `seekg` to the end of the stream, therefore determining the size of the file.

Data races

Modifies the stream object.
Concurrent access to the same stream object may cause data races.

Exception safety

Basic guarantee: if an exception is thrown, the object is in a valid state.
It throws an exception of member type `failure` if the resulting *error state flag* is not `goodbit` and member `exceptions` was set to throw for that state.
Any exception thrown by an internal operation is caught and handled by the function, setting `badbit`. If `badbit` was set on the last call to `exceptions`, the function rethrows the caught exception.

See also

istream::seekg	Set position in input sequence (public member function)
ostream::tellp	Get position in output sequence (public member function)