

File Streams

Solutions

- Explain the similarities and differences between `iostream` and `fstream`
 - `iostream` is used for input from and output to the system console
 - The C++ standard library provides unique instances for this (`cin` and `cout`)
 - `fstream` is used for reading from and writing to files
 - There are many possible file streams, so we have to create the instances ourselves

- Describe how to open a file for reading (i.e., receiving input from the file)
 - We can pass the file's name as argument to the `fstream` constructor
- How can we tell whether the file was successfully opened?
 - `fstream` has a `bool` operator. This will return `true` if the file was successfully opened, `false` if it was not

- Write some sample code which opens a file and reads text from it, one word at a time
- Are there any disadvantages to reading it this way?
 - Discards all whitespace (may not be what is required)
 - Difficult to handle unexpected input
 - Difficult to process or validate input

- Convert your code to a full program (remember to include `fstream`). You will need to put the input text file in the same directory that the program will be executed in
 - For an IDE, this can usually be done by creating a new file in the project

- Write some sample code which opens a file and reads text from it, one line at a time, into a `std::string` variable
- What data will be stored in the variable after each line is read?
 - The variable will contain the entire line of text, except for the newline character at the end
- Convert your code to a full program as described in the previous slide

- Write some sample code which opens a file and writes some text to it. Call the output file `hello_out.txt` to avoid overwriting the other file
- Verify that the output file `hello_out.txt` has been created and contains the correct text.
 - The file will usually be in the same directory that the program runs in. If you are using an IDE, you may need to check the project settings to find where this is

- When we call `close()` on an `ofstream` and there is still data in the buffer, what happens to it?
 - The buffer will be “flushed” by sending the data to the file
- What happens if we forget to call `close()` on an `fstream`?
 - The `fstream` will be closed automatically at the end of the program
 - If a program opens too many files without closing them, it may use up its allocation of open files. In this case, the operating system will not allow it to open any more files