

Stream Member Functions and State Solutions

- Explain what the open() member of fstream does

It will prepare the file for use and bind it to the stream. For ofstream, it will apply the file mode and create the file if it does not already exist

- Explain what the is_open() member of fstream does

Returns true or false depending on whether the file is open

Which of the following statements are true?

- A) By default, files are opened in text mode
- B) If we write 42 to a file, it will be stored as the number 42
- C) By default, output files are opened in "append" mode
- D) We pass the file mode as an optional second argument when opening the file

A and D are true

- Give an example of how to open a file in append mode
`ofile.open("hello.txt", ofstream::append);`
- What effect does this have? How does this differ from the usual way of opening a file?

Instead of overwriting the existing data, the data is added after the end of the file

- Create two files myfile1.txt and myfile2.txt (to follow your convention) which both contain some text, say, "abc"
- Write a program which
 - Opens myfile1.txt in the default mode and writes some text, such as "def" to it
 - Opens myfile2.txt in append mode and writes some text to it

- Run the program and compare the two files. Explain the results
 - myfile1.txt contains “def”
 - Opening the file in default mode causes the existing contents of the file to be deleted
 - myfile2.txt contains “abcdef”
 - Opening the file in append mode preserves the existing contents of the file

- How does binary mode differ from the default text mode?

Data in the file is handled as binary numbers instead of text. The `read()` and `write()` member functions have to be used instead of the `<<` and `>>` operators

- Give some of the drawbacks of using binary mode

Low-level, complicated and error-prone

- Give an example where binary mode is needed

Where the data is best represented by binary numbers, e.g. media files

Stream State Member Functions

- Name two member functions that can be used to check the state of a stream

`good()`, `fail()`, `bad()`