CSE-4301
Object Oriented Programming
2022-2023

Week-10

Stream and Files

-Faisal Hussain
Department Of Computer Science And Engineering, IUT

Contents

- Disk Files I/O with Streams
- File Pointer
- Error handling in File I/O

Disk File I/O with Streams

- ifstream – For taking Input
- fstream – For Input and Output both
- ofstream – For Output

- They are declared in <fstream> header file

C style file I/O

- fread()
- fwrite()

… will still works in c++

…be careful about mixing the old C functions with C++ streams. They don't always work together gracefully, although there are ways to make them cooperate.

Writing Data

- ofstream out(“filename”);

- various resources for the file, and accesses or *opens* the file of that name on the disk.
- If the file is not present, one will be created at the beginning.

- out<<“Hello”;

- When the program terminates, the out object goes out of scope. This calls its destructor, which closes the file, No need to close the file explicitly.

Writing Data

- Since numbers are stored as a sequence of characters, rather than as a fixed-length field, you must separate numbers with nonnumeric characters.

Reading Data

- ifstream in(“filename”);

- Blanks in string will be assumed as different string.
- char ch; int j; double d; string str1,str2;
- in >> ch >> j >> d >> str1 >> str2;

- Use getline() to read one line at a time.

Reading from File

- Detecting end of file – eof() member function
- The EOF is a signal sent to the program from the operating system when there is no more data to read

- while(infile.good()) // until any error encountered
- Using ios error flags

- while(infile) // until any error encountered
- If everything is going well, the object returns a nonzero value

Character I/O

- put() and get() functions, which are members of ostream and istream can be used to output and input single character.

Binary I/O

- To store large amount of numerical data it is more efficient to use binary I/O.
- In this form data is stored as like RAM.
- In binary I/O int -> 4 bytes. But text version “12345” -> 5 bytes

- ofstream os(“edata.dat”, ios::binary);
- For Binary I/O we will use
- write(), a member of ofstream; and
- read(), a member of ifstream
  - parameters -> address of the data buffer and its length.
  - address must be cast, using reinterpret_cast, to type char*,

Object I/O

- Writing an object, generally use binary mode, write().
- This writes the same bit configuration to disk that was stored in memory, and ensures that numerical data contained in objects is handled properly

- Reading an object back, use read() [file is in binary mode]

Acknowledgement

- Ashraful Alam
  Assistant Professor,
  Department of Computer Science Engineering
  Islamic University of Technology