COEN-244 Tutorial #9

March 23rd, 2023

C++ Input and Output

In C++, I/O libraries enhance the programming experience for handling input and output streams in different formats.

- C++ I/O occurs in streams, which are sequences of bytes

I/O Libraries

- **`<iostream>`** : Standard I/O library that provides the basics services for I/O operations
- `<iomanip>` : I/O Manipulating library that uses manipulators for formatted I/O
- **`<fstream>`** : Standard File I/O library that handles reading and writing streams to files
- `<sstream>` : String Stream library that associates a string object with a stream

<IOSTREAM>

The **`<iostream>`** header defines the **`cin`**, **`cout`**, **`cerr`** and **`clog`** **objects**:

- **`cout:`** The standard input stream
- **`cin:`** The standard output stream
- `cerr:` The unbuffered standard error stream and
- `clog:` The buffered standard error stream

POP QUESTION: Are these functions, objects, or pre-defined datatypes?

- These are pre-defined objects that are "connected" to standard I/O devices

COUT functionalities: insertion operator (**`<<`**); output of characters via (**`put`**); unformatted output via (**`write`**); formatted output (**`dec, hex, oct, precision …etc.`**)

CIN functionalities: extraction operator (**`>>`**); character input via (**`get`**); line of characters input via (**`getline`**); End-of-line (**`eof`**); input with ignore via (**`ignore`**); copied-stream input via (**`peek`**); input with text-insert at the previous location via (**`putback`**)

<FSTREAM>

File I/O is for **data persistence** (permanent storage of the data).

- The **`<Fstream>`** header includes stream communication channels to files.

<fstream> includes:

- `ifstream` class – for file input only
- `ofstream` class – for file output only
- `fstream` class – for file input and output

For an `ofstream` object there is two file-open modes:

  - **`ios::out`**: output data with over-write
  - **`ios::app`**: output data by appending - keep the original contents of file

For an `ifstream` object there is one file-open mode:

  - **`ios::in`**: input-only mode – unintentional modification cannot be made

File-Position Pointers

- Programs normally read from and write at the beginning of a file.

`<iostream>` library provides the member functions:

- `seekg` – re-positions the pointer for get
- `seekp` – re-positions the pointer for put

Arguments for these methods:

- Arg1 (int): relative byte number from relative position
- Arg2 (ios): relative position – `ios::beg`, `ios::cur`, `ios::end`

Exercise

Exercise 1: `SortMyFile` class

The class is supposed to do the following:

1. Read from `‘unsorted.txt’` and store it in a string
2. Convert the string into an array of numbers
3. Make a sorting method *(Choose a Sorting Algorithm)*
4. Invoke your sorting algorithm on the data
5. Return the sorted array
6. Write the sorted data to a file called `‘sorted.txt’`

Bonus: Integrate some exception handling!

THANK YOU