

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



## fstream class and open()

---

- ▶ `fstream` class is derived from `iostream` class
- ▶ `fstream` object can take both input and output.
- ▶ `open(file_name,mode)` function can be used to open a file. First parameter is a string and second parameter is the mode of opening.
- ▶ The vertical bars between the flags cause the bits representing these flags to be logically combined into a single integer, so that several flags can apply simultaneously.



# Mode bit

<i>Mode Bit</i>	<i>Result</i>
in	Open for reading (default for ifstream)
out	Open for writing (default for ofstream)
ate	Start reading or writing at end of file (AT End)
app	Start writing at end of file (APPend)
trunc	Truncate file to zero length if it exists (TRUNCate)
nocreate	Error when opening if file does not already exist
noreplace	Error when opening for output if file already exists, unless ate or app is set
binary	Open file in binary (not text) mode

# File Pointer

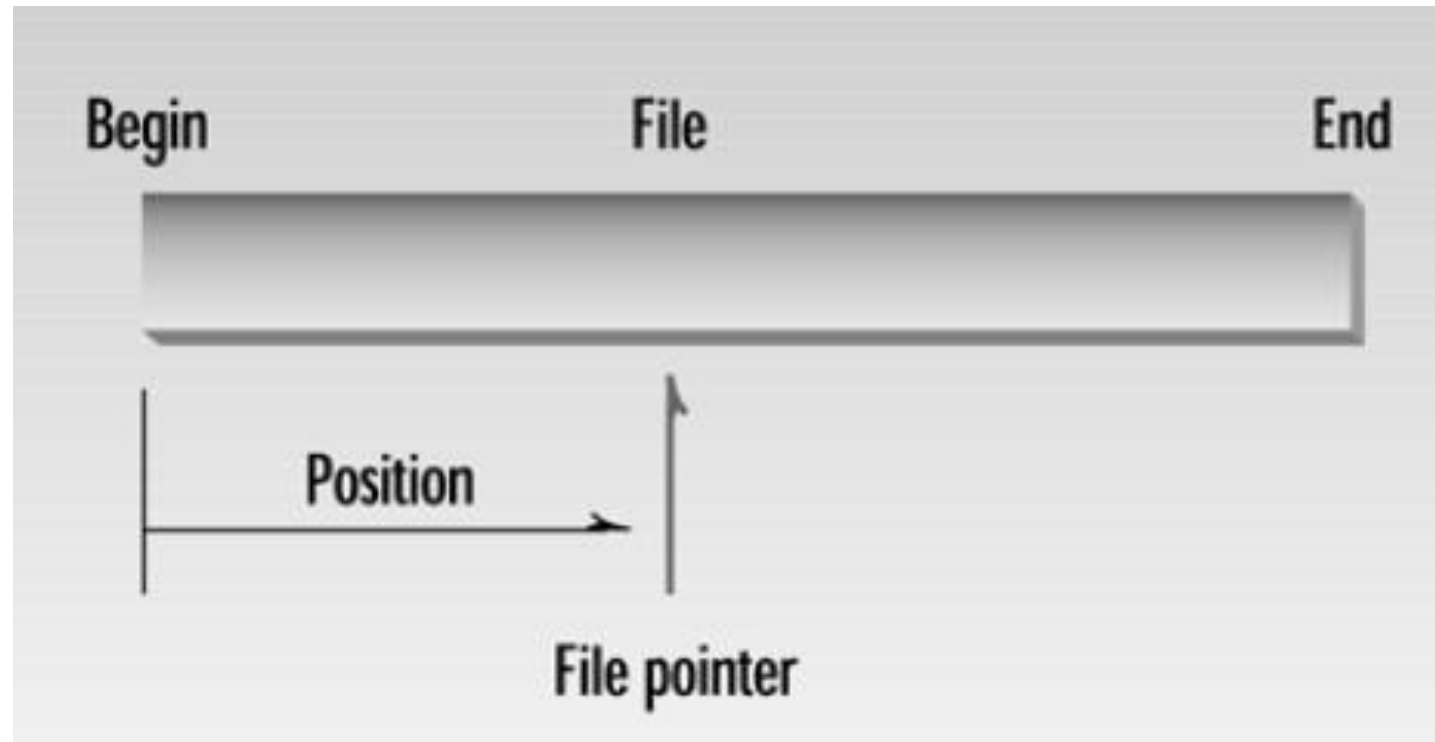
---

- ▶ File object has 2 integer values
  - ▶ *get pointer (current get position)*
  - ▶ *put pointer (current put position)*
- ▶ read from and write to arbitrary position
  - ▶ *Set and examine the get pointer*
    - ▶ *seekg()*
    - ▶ *tellg()*
  - ▶ *Set and examine the put pointer*
    - ▶ *seekp()*
    - ▶ *tellp()*



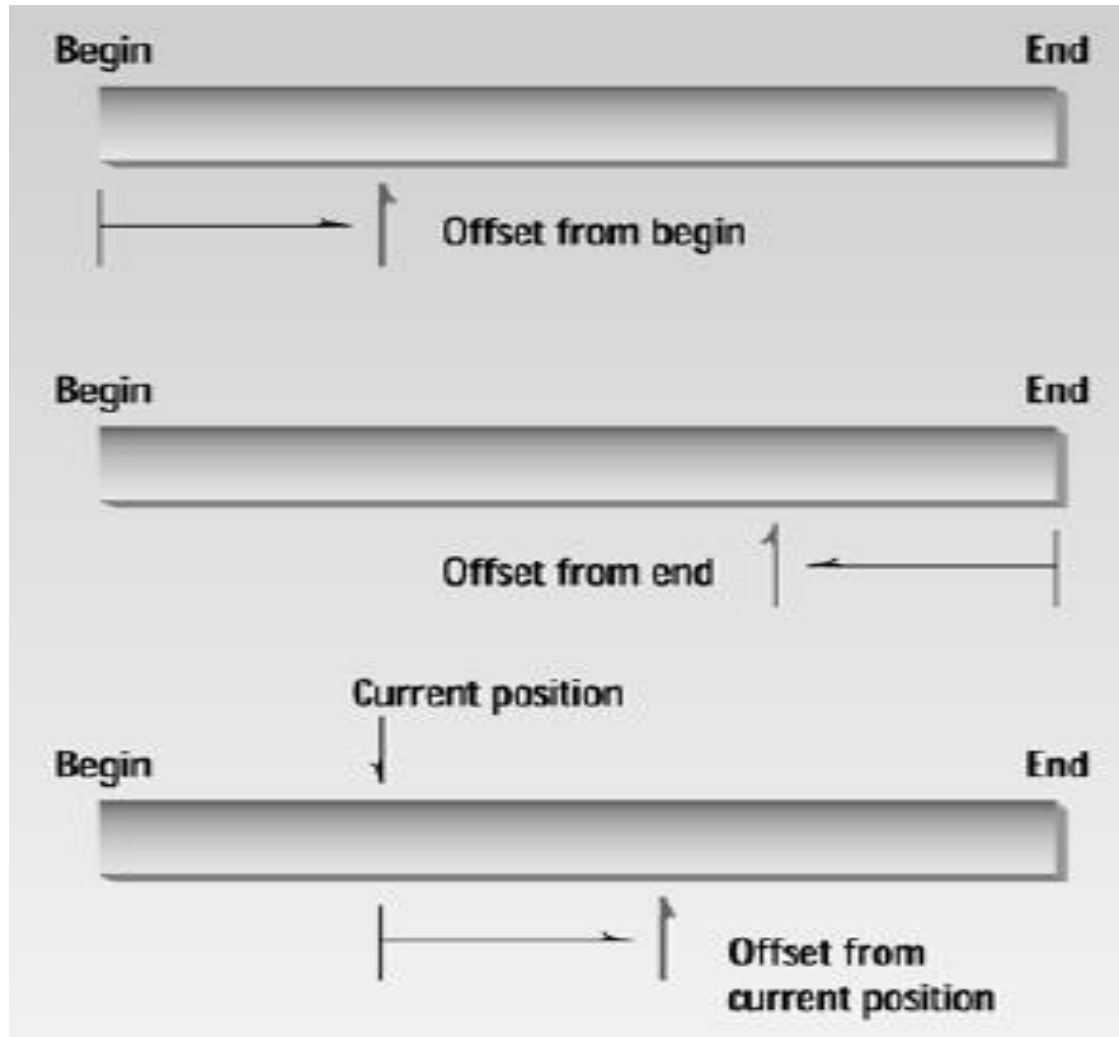
# Specifying the position

---



- ▶ `seekg(absolution_position_in_byte)`
- ▶ `seekg(0)` -> the start of the file is byte 0

# Specifying the Offset



- ▶ There are three possibilities for the second argument: **beg** is the beginning of the file (`ios::beg`), **cur** is the current pointer position (`ios::cur`), and **end** is the end of the file (`ios::end`).
- ▶ `seekg(-10, ios::end)`
- ▶ **`tellg()` returns the current position of get pointer**

# Error Handling in File I/O

---

- ▶ A file that you think exists may not,
- ▶ or a filename that you assume you can use for a new file may already apply to an existing file.
- ▶ Or there may be no more room on the disk, or no disk in the drive, and so on.





# Objects that Read and Write Themselves

---

- ▶ May be two member functions
  - ▶ `diskOut()` to write on file
  - ▶ `diskIn()` to read from file



# Classes that Read and Write Themselves

---

- ▶ Using static function
  - ▶ Write all objects using the static member function
  - ▶ It can access an array of pointers to the objects which can be stored as static data.



# Overloading the Extraction and Insertion Operators

---

- ▶ We can overload the extraction and insertion operators so they work with the display and keyboard (cout and cin) alone
- ▶ `friend istream& operator >> (istream& s, Distance& d)`
- ▶ `friend ostream& operator << (ostream& s, Distance& d)`



# Acknowledgement

---

▶ **Ashraful Alam**

Assistant Professor,  
Department of Computer Science Engineering  
Islamic University of Technology

