C++ Programming – Lecture 12

Input / Output

- Polymorphism Expectation from an IO System :
  - I should be able to communicate with sources & destinations
  - I should be able to I/O varied entities
  - I should be able to communicate in multiple ways
  - I should be able to deal with underlying file system
- C++ solution - Perform all IO using Streams
- Stream is a sequence of bytes that travel from source to destination over a communication path
- Streams are implemented by classes in iostream library
- Linking of Streams to physical devices is done by C++ IO system
- C++ program performs IO by reading / writing from / to a stream
- Benefits of using Streams
  - Streams hide details of communication from programmer
  - Methods are same, implementation changes as per device
- Ready-made stream objects :
  - cin, cout, cerr, clog - I/O of ASCII characters
  - cin - object of istream class
  - cout, cerr, clog - objects of ostream class
- There are many stream manipulators that help you manipulate the output as it is sent to an output device
- For manipulators that use arguments we should include the file 'iomanip'
- It is possible to create user-defined manipulators
- Using classes present in 'fstream' it is possible to perform 3 types of file I/O operations :
  - Character I/O - read/write file character by character
  - Line I/O - read/write file line by line
  - Record I/O - read/write file record by record
- For random-access in a file following functions are useful :
  - seekg(), seekp() - set get and put pointer in file
  - tellg(), tellp() - return current location of get and put pointer in a file
- There are multiple modes in which a file can be opened
- If file opening fails, then reason for failure is available in bits of a state variable