

# Agenda

---

- File i/o
- Manipulators
- Nested and local class
- Features of c++17
  - Nested namespace
  - const expr
- singleton Design Pattern

## Streams

---

### file i/o (demo01 to demo04)

---

### manipulators (demo05 and demo06)

---

- manipulators are the one which are used to manipulate the output and not the values.
- manipulators are of two types
  - 1. manipulator without arguments
  - 2. manipulator with arguments
- if we want to access the manipulators then we have to use the header file iomanip.

### Local class (demo07)

---

- a class defined inside the function is called as a local class
- we cannot create object of the local class outside the function.
- local class cannot access local variables of that function.
- however it can access global variables as well as static local variables of that function.

### Nested class (demo08)

---

- a class inside another class is called as nested class.
- in this we have an outer class and an inner class.
- data members of outer class are not accessible directly inside the inner class however they can be accessed using outer class object.
- static data members of the outer class are directly accessible inside the inner class.
- global variables are accessible directly inside the inner class.
- to create an object of nested classes we have to use name of the outer class scope resolution and name of the nested class.
- if the nested classes are kept private we cannot create its object outside the outer class.

# features of c++17(demo09 to demo12)

---

## singleton design pattern (demo13)

---

- Singleton design pattern is a way in which we create only one instance of our class.
- we cannot create multiple instances.
- To create this design pattern make the ctor as private.
- provide a static function (getInstance()) which will return an object of that singleton class.
- create a static member of pointer type of the same class.
- keep the initial value inside the ptr as NULL;
- inside the getInstance() check for if the ptr is holding any object of the singleton class.
- if already the object is created return the same object, else create a new object and then return it.