Agenda

```
- History
- Limitation of C
- OOP concepts
    - Major Pillars
    - Minor Pillars
- Datatypes in CPP
- Structure, in c and CPP
- Access Specifiers in Structure
- Namespace
- cin,cout
- Function Overloading
```

History

- Bjrane Stroustrup
- C With Classes
- CPP OR C++

Limitation of C

- POP -> Procedure Oriented Programming
- As the code size increases it becomes complex to maintain such code
- Their is o security for your data

OOP -> Object Oriented Programming

- It is a methodology
- If any programmaing language follows this methodology then we call such programming languages as OOP languages
- Abstraction,Encapsulation,hirerachy,ploymorphism
- Major Pillar
    - Abstraction
    - Encapsulation
    - Modularity
    - Hirerachy
- Minor Pillar
    - Typing/Polymorphism
    - Persistance
    - Concurrency
- following all the major pillars is compulsary for any programming language that term itself as OOP Language

Abstraction

- knowing only the essential things
- eg-> call given to the function
- printf(), scanf()
- projector,mobile

Encapsulation

- Binding the data and code together
- defining the functions,structure,class is called as encapsulation
- abstraction is an outcome of encapsulation

Modularity

- writing the code in different differnt modules
- Multilple function or Multiple files

Hirerachy

- examples
- OS
- intel Processors

Polymorphism/Typing

- An entity that takes multiple different forms is called as polymorphism
- eg -> Mobile, WhiteBoard

Persistance

- Storage
- to persist the data permanantly
- eg ->File i/o

Concurrency

- The execution should be concurrent

Hello World (demo01.cpp)

- give the extension to the file as .cpp
- include the header file
- write the entry point function main same as 'C'
- Start the programming..😃
- to execute the program
    - g++ demo01.cpp
    - a.exe

Datatypes in CPP (demo02.cpp)

- bool -> boolean -> 1 byte
- wchar_t -> wide character -> 2 bytes

Structure, in C (demo03.c)

- we cannot write functions inside structure

Structure, in CPP (demo03.cpp)

- We can write functions inside structure

Access Specifier in structure(demo03.cpp)

- public
    - All the members that are public will be accessiable directly on the object outsude the structure
- private
    - All the members that are private will not be accessiable out side the structure
- By default members of the structure are public

Namespace (demo04.cpp to demo08.cpp)

- It is a container that can store variables, functions,structure,classes
- We cannot cretae object of the namespace
- To access the members of the namesapce we have to use an operator called as Scope Resolution Operator ::
- <namespace_name>::

cin & cout (demo09.cpp)

- cin
    - It is an object of istream class
    - we need to use extraction(>>) operator with cin

- cout
    - It is an object of ostream class
    - we need to use insertion(<<) operator with cout

Function Overloading (demo10.cpp)

- It is an example of Compile Time Polymorphism
- To overload the function we need to keep the name of the function same but the signature should be different
- No of parameters should be different
- Or the type of parameters should be different
- if the no and types are same then their order of parameters should be different