Agenda

- Access specifier in structure
- cin & cout
- Function Overloading
- Default Argument
- Inline Function
- Class
- Object
- Access Specifier in class
- ~~this pointer~~
- ~~Namespace~~

Access specifier in structure (demo)

1. private
    - it provides accessiability only within the struct and not outside the struct
    - variables inside the structure are made as private
2. public
    - it provides the accessiability within as well as oustide the structure
    - functions declared inside the structure are made as public

- By default the memebrs of the structure are public.

cin and cout (demo01)

- cin is an external object of istream class
- cout is an external object of ostream class
- both these objects are defined inside a namespace called as std in the iostream header file.
- cin is used to take input from user (simialr to scanf in c)
- cout is used to display the output on console (similar to printf in c)
- with cin we have to use an operator called as extraction operator (>>)
- with cout we have to use an operator called as insertion operator (<<)

Function Overloading (demo02 and demo03)

- definining multiple functions with same name and differnt signature is called as function overloading
- it is an example of compile time polymorphism
- what does this different signature means

1. No of paramaters
2. type of parameters
3. order of type of parameters

- return type is not considered in the function overloading

Default Argument Function (demo04)

- for a function parameter we can alos provide default values
- such functions are called as default argument functions
- in default argument function the default values must be given from the rightmost parameter.

Inline Function (demo05)

- inline is a keyword in cpp
- it is used to request the compiler to resolve the function by copying the function at the function call position.
- It is used only when the functions are smaller and the execution time of function is less then the function overhead.
- It is just a request that is made to the compiler which can get rejected if the functions are big.

Class

- class is called as a logical entity
- It is used to bind data and code together.
- It is an example of encapsulation
- It is also called as an blueprint of an object
- class consists of datamembers(varaiables) and member functions(functions)

Object

- object is physical entity
- It is an instance of a class
- object defines 3 things

1. state
    - datamembers of the class determines the state of an object
2. behaviour
    - member functions of the class determines the behaviour of an object
3. identity
    - the values inside the datamembers is used to identify the objects uniquely
    - if the state of the objects is same we can use the address to identify the object

Access Specifier in class

1. private
    - members are accessiable within that class and not outside the class
    - datamembers should be kept as private
2. protected
    - members are accessiable witin the class and inside the derived classes.
    - however they are not accessiable outside the class.

- We will learn it at the time of inheritance

3. public
    - memebrs are accessiable within,outside the class
    - member functions must be made as public.