Agenda

- Object and its characteristics
- Scope Resolution Operator
- Namespace concept and its use
- Console streams
- Inline Functions
- Function Overloading
    - Name Mangling
- Default Argument Function
- Menu driven code

class (demo01)

- It is an entity that binds data and code together
- It is an logical entity
- It is also called as blueprint of an object

object (demo01)

- It is a physical entity
- It defines 3 things
    1. state
        - data members of the class represents staate of an object
    2. Behaviour
        - member functions of the class represents behaviour of an object
    3. Identity
        - the unique data members of the class represents the identity of an object.
        - If the class does not have unique data member then the address of an object represents identity

Scope Resolution Operator (:😀 (demo02.cpp)

- It is used to access the members of the namespace
- To access the global variables we need to use the operator along with the member name.

namespace (demo03 to demo08)

- It is a container used to aoid name ambugity/collisions
- It is also used to organize the code.
- To access the members of the namespace we have to use the name of the namespace followed by scope resolution operator and then the member name

    - eg -> namespace::

Streams (demo09 and demo10)

- istream
    - cin is an exten objects of istream class
    - cin is a part of std namespace
    - cin object uses extraction operator(>>) to take input from console.
- ostream
    - cout is an exten objects of ostream class
    - cout is a part of std namespace
    - cout object uses insertion operator(<<>) to display output on console.

Inline Functions (demo11)

- It is request made to the compiler to resolve the function at compile time.
- The functions that are made as inline takes time to get resolved as the contents get copied from that function in the function where they are called.
- this increases the compilation time.
- making functions inline will reduce the execution time.
- however as it a request made to the compiler their is no gurantee of geting the function resolved at compile time.
- if the functions are small with min no of statements with som vlaues they are retuning, such functions have high possibility of getting resolved at compile time.

Function Overloading (demo12)

- Defining multiple functions with same name but differnt signature
- To overload the function the signature can vary as below
    - to overload the function the no of parameters should be differnt
    - if no of parameters are same then the type of parameters should be different
    - if no and types are same then theor order should be different
- It is an example of compile time polymorphism
- Name Mangling
    - When conpiler adds an extra information to the functions depending on the paramters that are passed, such process is called as name mangling.

Menu driven code (demo13)

Default Argument Function (demo14)

- If depending on no of parameters that are passed their is no any major change in business logic then instead of function overloading we can go for default argument function.
- if we want to pass multiple parameters to the functions and want to have default values for some of its parameters then we can use default argument function.
- the default values to such functions should be given from the rightmost parameter.