Agenda

- Inline function
- Function Overloading
- Default Argument Function
- class
- Object
- Data Members
- Member Functions
- Access Specifiers
- this pointer

Inline Function (demo01.cpp)

- It is just a request made towards the compiler to resolve the function at compile time
- The request can be accepted or rejected depends on the compiler wether it wants to resolve it or not
- Thr inline functions if gets resolved at compile time then they helps to execute the code a bit fast.
- but the time of compilation increases

Function Overloading (demo02.cpp)

- Writing function with same name but with different signature
- differnt signature means ->
    - 1. change in the no of paramaters
    - 2. Change in the type of parameters
    - 3. Change in the order/sequence of parameter types
- Function overloading is an example of compile time polymorphism
- return type does not matter for fucntion overloading
- it is possible in cpp because of a concept called as name mangling

Default Argument Function(demo03.cpp)

- A function that has default values to its paramters.
- If the values to the function is passed at the time of function call for these parameters then the given values will be considered.
- If the values to the function are not passed at the time of function call for these parameters then the default value assignd to them are considered
- the default values shoulld be assigned from the rightmost parameter of the function

class (demo04.cpp)

- class is a logical entity

- class encapsulates data and code together
- class is also called as blueprint of an object

Object (demo04.cpp)

- Object is a physical entity
- Inside object the memory will be assigned only to the datamembers of the class
- Object determines 3 things
    - 1. state
    - 2. Behaviour
    - 3. Identity

DataMembers (demo04.cpp)

Member Functions (demo04.cpp)

1. Constructr
2. Destructor
3. Mutators
4. Inspectors
5. Facilitators

Access Specifiers (demo04.cpp)

- private
- public
- protected
    - protected access specifier we will be studying at the time of inheritance
- By default the members of class are private

this pointer(demo05.cpp)

- this pointer is a const pointer passed intrnally to all the member functions of the class.
- this pointer is of same type as that of your class
- if you want you can access the members of the class using this pointer.
- using this pointer to access the members is optional but it is industry standard practice to use this pointer.