# Variables

**Instance variable :-**

(maintain multiple copies for each instance of the class created. Initialized only when instance created **N** instances **N** copies **0** instances **0** copies)

**Static :-**

Maintain one value for the whole class can be modified but will not create multiple copies initialized one and only one time

**Const :-**

These variable can’t be modified after declaration must assign the value at the time of declaration can’t modified maintain one and only one copie through out life cycle of the class.

**Readonly :-**

These variables also can’t be modified after the initialization (after declaration maintain **N** instances **N** copies **0** instances **0** copies)

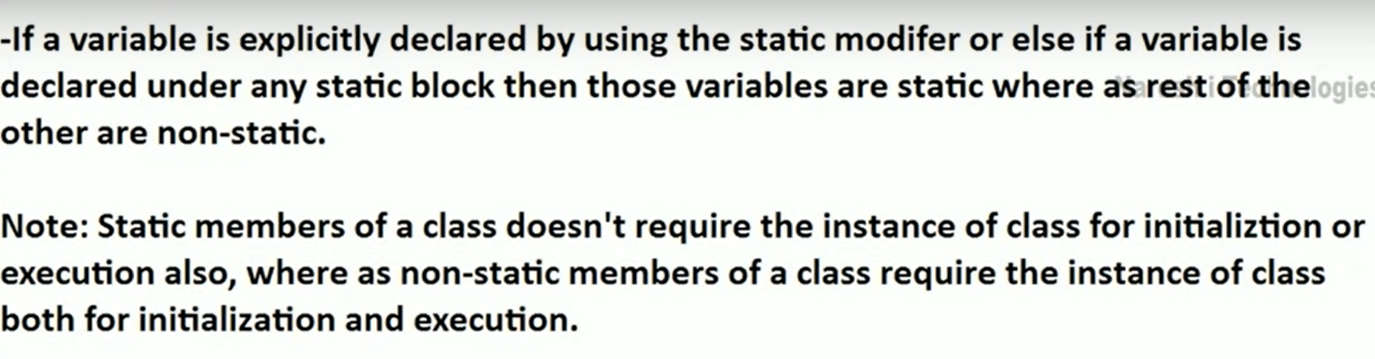
**Modifiable Non-Modifiable:-**

**Static :-** single copie modifiable static

**Constant** single copie non modifiable static

**Instance:**-Multiple copies modifiable

**ReadOnly** :- Multiple copies non modifiable



Static variables of a class are initialized immediately once the execution of class starts where as instance variables are initialized only after creating the class instance as well as each and every time the instance of class is created

If a variable is declared by using the keyword “const” we call it as a constant variable and these constant variable can’t be modified once after their declaration, so it’s must to initialize constant variable at the time of declaration only.

Const float pi=3.14f;

The behaviour of constant variable will be similar to the behaviour of static variable i.e. initialized one and only one time in the life cycle of a class and doesn’t require the instance of class for accessing or initializing.

The only difference between a static and constant variable is static variable can be modified where as constant variable can’t be modified.

If a variable is declared by using the readonly keyword we call that variable as a readonly variable and these variables also can’t be modified like constant but after initialization. It’s not compulsory to initialize a readonly variable at the time of declaration, they can be initialized under the constructor

The behaviour of readonly variable will be similar to the behaviour of non-static variables, i.e initialized only after creating the instance of class and once for each instance of the class created.

The only difference between readonly and instance variable is instance variable can be modified but not readonly variable.

Constant variable is a fixed value for the whole class where as readonly variable is a fixed value specific to an instance of class.

# Inheritance

It’s a mechanism of consuming the members of one class in another class by establishing parent/child relationship between the classes which provides reusability

How to perform the inheritance Consuming the members of A from B

Class A

{

Members

}

Class B:Class A

{

Consuming the members of A from B

}

Consuming the members of A from B

(A=>Parent or Base or Super)

(B=>Child or Derived or Sub)

Note:- Child class can consume members of it’s parent class as if it is the owner of those members(Except private members of parent of parent).

child can Inherit all the (business/properties) of my father but not job (because he get his job based on his qualification)parent qualification and child qualification not same

Copy and paste the code i.e(copy parent class methods into child class it’s not reusability reimplementing the code) Impact the size of the application(let us perform the inheritance)

**Important Points:-**

* Parent classes constructor must be accessible to child class, otherwise inheritance will not be possible

Class1 and class 2 both contains implicit constructor(every class contains implicit constructor if not define explicitly)

If a constructor defined implicitly by default public

If I defined class1 Explicit constructor whenever child class instance created it will call parent class explicit constructor

Private constructor inaccessible due to protection level

* In inheritance child class can access parent classes members but parent classes can never access any member that is purely defined under the child class.
* We can initialize a parent classes variable by using the child class instance to make it as a reference, so that the reference will be consuming the memory of child class instance. But in this case also we can’t call any pure child class members by using the reference.