**variable**

**data-type (string, int, char, Boolean, float)**

**operator**

**condition (if, else-if, switch)**

**loops (for, while, do...while, foreach)**

**break, continue**

**function (default / parametrized)**

**access modifiers**

**call by value / call by parameter**

**arrays**

**class / object**

**constructor / destructor**

**inheritance**

**operator overloading**

**operator overriding**

**polymorphism**

**abstract**

**Enum**

**Generics**

**. collection (namespace)[list<t>, hashset<t>, sortedset<t>, stack<t>, queue<T>, linklist<T>, Dictionary <TKey, TValue>,** **SortedDictionary** **<TKey, TValue>]**

**LINQ (query expression / query method)**

##### ===========================================================

**Variable**

**Datatype (int, string, char])**

**Class** (it is blueprint of an object)

**Object** (run time entity / instance [ANY TYPE OF VAR.] of a class)

**abstraction** (common filed ne ek sathe call karva)

**inheritance** (parent /child class)

**polymorphism** (teacher behave school, home party etc)

**constructor** (default / parameterized)

**method overriding** (virtual class ne override karva mate)

**method overloading** (multiple method with the same name)

**interface** (it is a method and class ma obj through work thay / blue print of class / abstract hoy)

**boxing** (object refer kare value ne / var mathi object ma value store kare)

**unboxing** (object mathi var ma store kare)

**list** (multiple value store kare)

**Enum Enumerator** (one type of array / return value index number

**Explicitly type** (data type define Kari ne thay)

**Implicitly type** (var define Kari ne thay) var j = 100;  
**Encapsulation** (binding the data members and member functions into a single unit. / e.g., any class)

**Struts** (struct can be used to hold small data values / define struts keyword)

**Generics** (allows us to define classes and methods / we must use angle <> brackets)

**List<t>** (list create and obj, through work kare)