Strings

Strings can be created or accessed like a char-array
Multiple strings can be connected with a ~~f~~ "+"

Generics - not ~~a~~ specific to a particular datatype
generic type is declared by specifying a type parameter in an angle brackets after a type name: TypeName<T>

Generic class
- ~~ne~~ increases reusability
- can be a base class to other generic or ~~un~~ non-generic class
- can be derived from — " —

A Method declared with a type parameters for its own returtype or parameters is a generic Method
Generics are type safe, and have performance adventage because they remove the possibility of boxing / unboxing

Generic Collections

List<T> - contains elements of a specified type
Dictionary<TKey, TValue> - contains key-value pairs
SortedList<TKey, TValue> — " — " —. Adds elements in ascending order of key
Queue<T> - stores values FIFO style. Enqueue() method for adding values and dequeue() to retrive values.

Steck<T> - stores velues ~~FIFO~~ LIFO style. Push() to add velaes and Pop() and Peek() to retrive

Heshset<T> - List elements, eliminating duplicats

Non-Generic Colections

ArreyList - stores objects of eny kind but there is no need to specify the ~~[illegible]~~ size of the list as it grows automaticly

Heshtable ~~<~~ - stores key-velue pairs. retrives velues by compering hesh velues of the keys

BitArrey - meneges en arrey of bit velues (1 or 0) which are represented as a boolean

C# also Includes non-generic versions of Queue, ~~[illegible]~~ Steck and SortedList

Tuple<T> - dete structure thet conteins a sequence of elemets of different deletypes. its used when we went to hold en objet with properties but dont went to creat a seprete type for it. ~~[illegible]~~ the element can be accessed vie Item<elementNumber> or Rest if its the lest item. If we went to store more then 8 elements in a tuple, we can do thet by nesting another tuple objet in the 8th element. we can occess it via Rest.Item1.Item<elementNumber> In theory we can nest tuple object anywhere in the sequence but we won't be eble to acess its elements

Tuple is usefull when:
- we went to return multiple values from a method
- we wat to pess multiple velues to a method with a slugle peremeter
- we went to hold a detabese temporeirly witaut creating seperete ~~ct~~ class

STRUCTURE

struct - is a velue detatype that represent data structers. It can contein: perameterized or static constructor, constents, fields, methods, properties indexers, operetors, events and nested types.
It can be used to hold data that does not require inheritence. It can be declered using „new" operetor, if you don't do that it doesn't call any constructor so all members remein unassigned so you have to assign them.

~~struct~~ struct member cannot be specified as abstract, sealed, virtual or protected