Class UML Diagrammer:

- Et diagram hvor man kan se alle objekter i projekt.
- I hver objekt kan man se dens fields/propeties samt functioner/methods.

- Kan bruges til at dokumentere sine projekter.

Dictionaries:

- En objekt som indeholder et collection af key value pairs f.eks: Dictionary<type1>, <type2>
- Er dynamisk i størrelse
- Kan bruges til at lave relationer mellem objekter, hvor objekterne ikke behøver at vide noget om hindanden.

Overloading:

- En række Metoder som deler om samme navn.
- Hver metode skal have et unik signatur. Man kan ikke have flere med Method1(string, string)
- Kan bruges hvis man vil lave et funktion, som tager imod flere forskellig slags parametre.

Delegates:

- En type som peger på et funktion(lambda expression, eller metode).
- Typen indeholder return type og parametre.
  F.eks: public delegate string Delegate1(int num1)
- Kan bruges når man skal køre funktioner på tværs objekter, uden at kreere relationioner med hinanden.

Lambda Expressions:

- En måde at skrive et funktion til delegates, uden at pege på et metode i et klasse.
- Skal overholde et delegate signatur. F.eks
  public delegate int MyDelegate(string text);
  MyDelegate myDelegate = **(x) => x.Length;**
- Delegaten kan så kaldes med variablen 'myDelegate': int test = myDelegate("Abekat");
  myDelegate skulle så gerne have returnet: 6

Nedarvning:

- Properties/Fields og metoder kan nedarves fra et andet klasse.
- En klasse kan kun nedarve fra en klasse, men en klasse kan have flere som nedarver fra den.
- Ting som er 'private' bliver ikke nedarvet videre, men det gør 'protected'
- Metoder som er defineret som virtual, kan overrides af subclasses.
- Man kan arbejde med forskellige klasser, som alle nedarver fra samme sted. F.eks: Man kan have et liste med person, og fylde den op med person, samt klasser som nedarver fra person.

Abstract Classes og interfaces:

- Abstract class, er et class man kan nedarve fra, men ikke lave et instance fra.
- Abstract metode skal overrides af alle subclasses, som nedarver fra classen.
- Et Interface indeholder signature af metoder, som skal overrides af dem som skal bruger interfacet.
- Et class kan benytte flere interfaces.
- Et interface starter med et stort bogstav, og prefixes med et stort 'i'.
- Interfaces kan bruges, når man vil have de samme funktionaliteter, på tværs klasser. Uden at 'hard code' dem ind.

Library(dll.)

- Et class library, er et projekt som kreere et .dll fil, ud fra objekter.
- Man kan så fra andre projekter referere til denne dll fil, og kalde på dens objekter.