Viktig å vite – uke 7

*Mål: Objektorientering, klasser, konstruktører, innkapsling*

**Objektorientering:**

Objektorientering betyr at man lager objekter som inneholder data og metoder/funksjonalitet i programmer. Fordeler med dette er at vi kan fremstille gode modeller av den virkelige verden.

*Eks: En bil kan lages digitalt i et program hvor vi gir bilen dataen: et merke, et registreringsnummer, en farge og mer. Funksjonaliteten kan være metoder som at bilen kan kjøre, at man kan fylle bensin på bilen og mer.*

**Klasser:**

En klasse er en type objekt. Si for eksempel at en bil er en klasse og et menneske er en klasse. Hver klasse kan ha flere objekter laget utfra klassen. *Ta for eksempel klassen menneske. Hvert menneske har et navn, en alder og en høyde. Menneske er klassen. Ole er et menneske. Han er 19 år og er 179 cm høy. Ole er et objekt i klassen menneske.*

**Konstruktør**

En konstruktør bestemmer hvilke egenskaper/data et objekt skal ha når det blir opprettet. *Ta for eksempel klassen Menneske. Vi kan tenke oss at mennesker blir opprettet når det blir født. Dermed kan vi vite at mennesker skal være 0 år når det opprettes. I python har vi denne syntaxen for konstruktører:*

*class Mennesket:*

 *def \_\_init\_\_(self, navn, hoyde):*

 *self.\_navn = navn #self.\_navn er en instansevariabel*

 *self.\_hoyde = hoyde*

 *self.\_alder = 0 #Her setter vi alderen til 0 når et nytt -*

*#mennesket opprettes*

**Innkapsling**

Innkapsling betyr at vi kan bestemme hvilken data/instansevariabler og funksjonalitet/metoder vi vil at skal være tilgjengelig utenfor klassen/objektet. Dette bestemmer vi ved å gjøre noe i objektet privat og annet offentlig. Det brukeren av klassen skal bruke må være offentlig, alt annet bør være privat. I IN1000 skal alle instansevariabler (self.\_navn) være private. Dette gjøres ved å plassere en understrek etter self.(HER SKAL DET VÆRE UNDERSTREK)(variabelnavn).

*I eksempelet for konstruktøren brukes innkapsling av instansevariablene.*