

**Class:** Klasse er en blueprint på hvordan programmet skal være. Dette vil da kunne bestå av flere felt, metoder og forskjellige typer konstruktører. Ved definering av klasser som *public class Main {}*, vil class bety at innholdet i «{}» vil tilhøre Main-klassen.

**Objekt:** er en realisering / instanse av en klasse. Objektet kan inneholde metoder som bruker argumentene som er gitt til utføring av oppgaver.

**Instansvariabel:** Er variablene som blir deklartert i klassene, for hovedsakelig kun lokal bruk. Dette gir da en bedre flyt og forståelse for koden. Disse må da bli definert med public/privat/protected og med hvilken datatype som blir tatt i bruk.

**Overloading:** Er en funksjon som gir muligheten til å ha flere funksjoner med samme navn i en klasse. Det vil si at man kan lage flere metoder som krever f.eks. forskjellige antall av input argumenter som kreves for å gjennomføre metoden.

**Overriding:** Vil hovedsakelig si å overskrive noe. Dette gir da muligheten til å overskrive «innebygde» metoder som toString(). Dette gir utvikleren muligheten til å lage metoder med likt navn for de forskjellige klassene med forskjellige funksjoner.

**Extends:** Kan tolkes som å bygge videre på noe. I vårt tilfelle vil man «extende» en klasse med en annen. Det vil si at man «overfører» metoder osv til klassen som tar i bruk «extend» funksjonen. Dette gir da muligheten til å arve alle public metodene og variablene, noe som kan forenkle koden.

**Polymorphism:** Det vil si at alle objektene kan «utgi» seg for å være et objekt av alle klasser den arver fra. Det vil si at man får muligheten til å sammenligne/behandle en gruppe med objekter på likt grunnlag.

**Privat, public, (protected) (klasse, variabel, metode):** Privat, public er hovedsakelig tatt i bruk for å definere tilgjengeligheten til klasse/variabel/metode. Privat gir kun tilgang til variabelen eller metoden i den spesifikke klassen som brukes til å definere den. Public vil gi muligheten til å endre og bruke variabelen/metoden/klassen i et arvende eller videreførende klassen. Protected vil gi noe som ikke kan endres, det vil si at verdien vil være fastsatt.

**This og super:** This blir brukt til å få tilgang til metoder og variabler i den nåværende klassen og super blir brukt til å få tilgang til metoder og variabler fra foreldreklassen.