# C-Programmering

## 1: Verktøykjeden

Preprossesor setter inn verdier for makroene (defines, headerfiler etc.) og sjekker for syntaksfeil.

Kompilatoren tar .c filene og kompilerer dem til .s i assembly.

Assembleren gjør om .s filene fra assembly til maskinkode i .obj filer som maskinen forstår.

Linkeren tar .obj filene og setter dem sammen alle bibliotekene som programmet krever. Outputen er her en utførbar maskinkode.



## 2: Filer og moduler

1. En deklarasjon forteller kompilatoren at en funksjon vil eksistere og omtrent hvordan den vil oppføre seg (navn, return type, inngangstyper). En definisjon forteller alt det samme og også hva funksjonen faktisk gjør.
2. Det er ingen fundamental forskjell og man kan f. eks. skrive #include ‘’foo.c’’ og skrive hele programmet sitt i en ‘’main.h’’ fil, men da vil man få mange stygge blikk og ikke bli tatt veldig seriøst.
3. Inne i en funksjon betyr det at variabelen skal leve etter at funksjonen er ferdig, og at neste gang funksjonen kalles vil den ikke resettes.   
   En static global variabel vil kun være tilgjengelig i filen den er deklarert i. I en .c fil vil det si kun den filen, men dersom man deklarerer en static global variabel i en .h fil vil den være tilgjengelig (og lik selvsagt) i alle filer som inkluderer den filen.   
   En static global er egentlig nøyaktig det samme bare med en funksjon istedenfor en variabel.  
   Den interessante forskjellen mellom de tre tilfellene er at globale variabler er by default deklarert som static, mens funksjonsvariable og funksjoner ikke er det.
4. Får å få til delte variabler mellom funksjoner (tilstand og logg) trengs extern keywordet. «Use» får vi til med #include. Får å få til private funksjoner må static brukes. Selvsagt må også hver modul deles opp i .h og .c filer hvor de private funksjonene er deklartert med static i .h filene.

## 3: Warning & Errors

1. - error: ‘x’ undeclared (first use in this function)  
   Kompilatoren oppdager dette. Har nok noe med at x ikke er deklarert, men at man prøver å bruke den. Vet f. eks. ikke da hva slags type x har.  
   - warning: implicit declaration of function ‘foo’  
   Komplilatoren bruker ‘foo’ uten å vite deklarasjonen. Har altså laget en implisitt deklarasjon.   
   - undefined reference to ‘foo’  
   Lenkeren finner ikke symbolet ‘foo’. Kanskje hvis man glemmer å gi den en filen med ‘foo’  
   - fatal error: file.h: No such file or directory  
   Preprossessor som ikke finner header filen.
2. Segmentation fault er en feil som skjer når man prøver å aksessere minnet som ikke tilhører programmet. Operativsystemet vil da oppdage dette.

## 4: C-kode

1. sum blir ikke initialisert.
2. a = 8
3. Den får elementet som n\_first peker på til å bli det første elementet på en måte som beholder rekkefølgen til elementene.
4. Prøver å aksessere en peker som ikke har noen verdi, altså en udefinert minnelokasjon.
5. Den andre siden a i den først forsvinner når funksjonen er ferdig, mens i den andre ligger variablen på heapen.