

# Kommentering

For kommentering (og senere dokumentasjon) bruker vi Doxygen ( [www.doxygen.nl](http://www.doxygen.nl) ). Doxygen er en vel anerkjent og de facto standard for å genere dokumentasjon ut fra kommentarer/anmerkninger i C++ program. Men mange andre programmeringsspråk støttes også, f.eks: C, C#, Objective-C, Java, Perl, Python, PHP og Fortran.

**NB: Læreboka bruker (dessverre) *ikke* denne standarden!**

## Kommentarblokker

Generelt skrives en kommentarblokk som:

```
/**
 * ..... Tekst .....
 * ..... Tekst .....
 */
```

(Det er alternative måter å skrive dette på. Men, vi skal bruke denne Javadoc style måten. Den single '\*' på de to mellomste linjene er det frivillig å skrive. Men, vi skal gjøre det.)

Ønsker man evt. at kommentaren er enda mer synlig/uthevet i koden, *kan* man skrive:

```
/**
 * ..... Tekst .....
 * ..... Tekst .....
 */
```

**(NB: Husk da de to stk '/' før to stk '\*' mot slutten av første linje!)**

## Kort og detaljert beskrivelse

Den første *setningen* (frem til første punktum) i en slik blokk er definert som en *kort kommentar/beskrivelse* av det som nedenfor kommer (f.eks. en fil/program, funksjon, struct, klasse). Evt. mer etterfølgende tekst utgjør den mer detaljerte/lengre beskrivelsen.

Doxygen er stor og omfattende, og dere vil lære/bruke mer i andre emner senere. Men i dette grunnleggende emnet skal vi lære (på en standard måte) å kommentere:

## 1. Programfil

```
/**
 * En kort en-setnings beskrivelse av filens innhold. Avsluttes med '.' (punktum).
 *                                     (frivillig å ha med blank linje eller ei)
 * Evt. en lengre og mer detaljert beskrivelse som kan gå over flere linjer.
 *                                     (frivillig å ha med blank linje eller ei)
 * @file      Filnavn                (NB1: '@' kan erstattes med '\')
 * @date      Dato for koding         (NB2: Alle disse fire @-feltene er frivillige)
 * @version   Versjonsnummer
 * @author    Koders navn
 */
```

## 2. Globale variable / const'er / datamedlemmer (primært struct/enum)

Står kommentaren *etter* (som er meget vanlig) den globale variabelen/const'en/  
struct'ens datamedlem, brukes `///  
(evt. /**< ..... */)`  
Står kommentaren på linjen *foran*, brukes `///  
(evt. /** ..... */)`  
*Kommentaren bør omtale variabelens/const'ens/datamedlemmets hensikt/hva brukes til.*

## 3. Enkeltprogramsetninger

*Primært brukes `//` på alle kommentarer inne i en funksjon/blokk (da ignoreres den av Doxygen). Skal kommentaren være med (som er sjeldent), brukes `///  
Typisk er å kommentere (ikke-trivielle) programsetninger, løkker (for, while og do) og tester (if - else, switch - case) - hva skjer/gjøres.`*

**Men, vær bevisst på at linjer/kommentarer ikke blir for lange. La det gjerne være max. 80-100 tegn pr.linje. For ellers kan det bli en slitsomt leselig koden, når lange linjer wrapper om til neste tekstlinje ved en utskrift.**

## 4. Funksjoner

```
/**
 * En kort en-setnings beskrivelse av funksjonen. Avsluttes med '.' (punktum).
 *                                     (frivillig å ha med blank linje eller ei)
 * Evt. en lengre og mer detaljert beskrivelse som kan gå over flere linjer.
 *                                     (frivillig å ha med blank linje eller ei)
 *
 * @param   Beskrivelse av 1.parameter
 * @param   .....
 * @param   Beskrivelse av N.parameter
 * @return  Beskrivelse av evt. returverdi
 * @see     Henvisning til en annen kode (funksjon/metode, klasse, variabel, fil, URL)
 */
```

## 5. Struct / Class

```
/**
 * En kort en-setnings beskrivelse av structen/klassen. Avsluttes med '.' (punktum).
 *                                     (frivillig å ha med blank linje eller ei)
 * Evt. en lengre og mer detaljert beskrivelse som kan gå over flere linjer.
 */
```

## Bortkommentere kode

For f.eks. midlertidig å bortkommentere flere linjer med kode (som Doxygen ignorerer helt), kan det brukes:

```
/* .....
   .....
   .....
   ..... */
```