

# Skriva Program på ett strukturerat sätt

# Intr. till “Programmering i Praktiken”

---

ABC/LP

- För de som vill veta mer om hantverket programmering rekommenderas starkt boken:
  - “*The Practice of Programming*” av Kernighan och Pike, Addison Wesley, 1999.
- En grundprincip
  - “*Kod läses ett otal gånger fler än den skrivs*”

# Kodkonventioner

---

*ABC/LP*

- Handlar om hur man namnger variabler, konstanter, funktioner, filer
- Hur man layout-mässigt organiserar sin programkod
- Några olika kodkonventioner:
  - Indian Hill: <http://ieng9.ucsd.edu/~cs30x/indhill-cstyle.html>
  - Gnu: [https://www.gnu.org/prep/standards/html\\_node/Writing-C.html](https://www.gnu.org/prep/standards/html_node/Writing-C.html)
  - Nasa: <http://homepages.inf.ed.ac.uk/dts/pm/Papers/nasa-c-style.pdf>
  - Coding conventions (Wikipedia):  
[https://en.wikipedia.org/wiki/Coding\\_conventions](https://en.wikipedia.org/wiki/Coding_conventions)
  - K&R, dvs. hur Kernighan och Ritchie skriver i sin bok

# Varför?

---

*ABC/LP*

- Öka läsbarheten av programkoden
- Öka förståbarheten av programkoden
- Underhåll
  - Programkod lever under kanske ett otal år och ändringar görs, tillägg, buggrättningar.
  - Om koden är lätt att läsa resp. förstå blir jobbet enklare

- Kommentarer
  - För varje metod, vad den gör, vilka parametrar och dess innebörd, vad som ev. returneras och dess mening
  - För krångliga delar
- Skrivsätt:
- `//` resten av raden tolkas som en kommentar (C++-sättet), kanske inte finns överallt
- `/*` kommentar text `*/`, kan sträcka sig över ett otal rader

- Multirad-kommentarer skrivs ofta som:

```
/*  
 * Here is a block comment.  
 * The comment text should be tabbed or spaced over uniformly.  
 * The opening slash-star and closing star-slash are alone on a line.  
 */
```

- Småkommentarer kan skrivas som:

```
if (a == EXCEPTION) {  
    b = TRUE;                /* special case */  
} else {  
    b = isprime(a);          /* works only for odd a */  
}
```

# Namngivning

---

*ABC/LP*

- Konstanter
  - Skrivs med VERSALER
- T ex:

```
#define MIN_KONSTANT 23

if (tal < MIN_KONSTANT) {
```
- Variabler och funktioner
  - Välj namn som beskriver vad variabler innebär och som beskriver vad funktion gör
  - Långa namn bestående av flera ord avgränsas orden med \_  
`min_lilla_variabel`

# Indentering

---

*ABC/LP*

- Kodinskjutning
  - DVS. för varje nytt block i koden skall raden skjutas in med ytterligare *X* antal blanka (*X* = ett tabbsteg, el. 4 blanka eller ...), t ex

```
int my_F() {  
    int a = 10;  
    if ( a < 3) {  
        /* ..... */  
    } else {  
        while ( a >= 0) {  
            /* ... */  
        }  
    }  
}
```

# Programmeringsstil

---

*ABC/LP*

- Försök sätta KLARHET framför fiffighet:
- Vilket är bäst?

```
return (hours < 24 && minutes < 60 && seconds < 60);
```

- eller

```
if (hours < 24 && minutes < 60 && seconds < 60) {  
    return true;  
} else {  
    return false;  
}
```

- De är ekvivalenta

- **indent**, ett program som hjälper en att indentera källkod och gör också om man så vill enkel omformattering
- Gör rätt från början med hjälp av en bra **programmeringseditor**, som automatiskt indenterar korrekt

# Summering

---

*ABC/LP*

- Oavsett vilken kodstandard man använder är **KONSISTENT** användning det viktigaste
- och
- **KISS**-principen

*“Keep It Simple Stupid”*