

# Inför laborationerna

## Översikt

Ni är här för att lära er, så lös uppgifterna på ett sätt som ni lär er på, inte bara för att lösa uppgiften.

### Ett sätt att arbeta

Så länge som uppgiften inte är löst:

Fundera på/Ta reda på hur *en del av* koden bör se ut, skriv ner den *på papper*.

Testa den i dator, modifiera tills den fungerar, *uppdatera på papper*.

Utöka med mer kod som behövs för att lösa uppgiften.

Renskriv lösningen, gör anteckningar om hur olika delar fungerar, och inte fungerar, och om annat som du upptäckt under tiden, *på papper*.

Tänk efter: Varför fungerade detta, varför funkade inte andra saker, hur funkar det, hur funkar det inte?

## Innan du börjar

Innan du börjar laborera är det viktigt att du läst igenom labbreglerna på denna sida. För att komma igång med själva programmeringen kan du bland annat läsa föreläsningsanteckningarna, kurslitteraturen samt testa exempel.

För varje labb är det naturligtvis viktigt att du läser igenom labbinstruktionen noggrant så att du förstår vad du ska göra. Om något är oklart så fråga.

## Laborationsregler

För godkänt på labbmomentet i kursen krävs att samtliga laborationer blivit godkända vid den muntliga redovisningen och att de lämnats in i på kursens webbplats.

## Laborationsinstruktioner och handledning

Labbinstruktioner finns tillgängliga här på kurswebbplatsen. Handledning, tips och annan hjälp med labbar får dels på varje schemalagt labbtillfälle, dels genom diskussionsforum på kursens webbplats där både lärare och studiekamrater kan komma till undsättning.

## Miljö

Följ muntliga instruktioner.

## Koden

Variabelnamn, och namn på funktioner ska väljas så att deras betydelse är tydlig. Följ befintliga konventioner. All programkod ska vara skriven med vederbörliga indragningar(indentering) och väl kommenterad. Stilen bör vara konsekvent och följa konventioner.

## Kommentarer

Alla filer ska vara försedda med kommentarer i början med följande information:

```
Ditt namn
Datum
Vilken kurs det gäller
Vilken labb det gäller
```

## Funktioner

Alla skapade funktioner ska kommenteras. Syftet med funktionen ska framgå. Det ska räcka att läsa kommentaren för att förstå *hur funktionen ska användas* (utan att sätta sig in i hela koden) *men inte hur den fungerar*. In- och utdata till funktioner måste kommenteras, det vill säga parametrar, returvärden och specialfall.

## Kodavsnitt

Det kan ibland vara lämpligt att använda en kommentar för att sätta en rubrik på ett kodavsnitt.

```
# Räknar värden
finns = 0
for( int v : varden )
    if(v == sokvarde)
        finns += 1
```

## Övrigt

*Kommentarer får inte fungera som ett manus under redovisningen!*

Kommentarerna *ska inte förklara hur själva språket fungerar*. Förutsättningen är att den som läser källkoden redan vet hur man programmerar. Kommentarer som t.ex. förklarar att en if-sats gör ett val och att en loop upprepar något ska *inte* vara med. *Den som redovisar ska själv veta sådant utan kommentarer.*

## Redovisning

Innan en laboration redovisas, kontrollera att den verkligen utför uppgiften enligt laborationsanvisningen.

## Inlämning

De flesta laborationsresultaten består av flera filer. Om det är fler filer än vad inlämningslådan tillåter, använd ett zip-verktyg för att packa filerna.