

# TDDC76 Programmering och datastrukturer

## Kursinnehåll

- Programspråket C++ och dess miljö
- Objektorienterad programutveckling
- Datastrukturer och algoritmer

## Mål

- Kunna programmera i C++, både procedurellt och objektorienterat
- Kunna använda grundläggande objektorienterad metodik – OOA, OOD, ...
- Förstå och kunna använda vanlig förekommande datastrukturer och algoritmer

## Förkunskaper

- Grundkurs i programmering väl inhämtad, dvs
  - grundläggande färdigheter i problemlösning och programmering
  - datorsystemvana (Unix, Emacs, ...)

## Påbyggnadskurser

Realtids- och processprogrammering, databaser, större projektkurser, avancerad programmering C++, ...

## Organisation

- Föreläsningar (28 h; 12 + 16): C++, OO, DoA
- Lektioner (11 h; 8 + 3): lab- och systemgenomgång, övning OOA, hemtenta
- Laborationer +projekt (60 h; 32 + 28): Laboration 1-3, miniprojekt (KS+OOA+OOD+OOP)
- Få föreläsningar och lektioner – mycket labbtid och hemarbete!

## Examination

- Laborationer (LAB2) – betygssätts U/3/4
- Projekt (PRA1) – betygssätts U/G
- Skriftlig hemtentamen (UPG2) – enbart DoA utan koppling till programspråk – betygssätts U/3/4
- Slutbetyget är ett sammanvägt sifferbetyg baserat på betygen på laborationerna och hemtentamen

## Kurslitteratur

- **C++ Primer**, 5Ed, Lippman/Lajoie/Moo – lärobok för nybörjare i C++
- **The C++ Programming Language**, (4Ed), Stroustrup – *referensbok, inte för nybörjare*
- **Kompendium om Datastrukturer och algoritmer** – finns på webben
- Det finns mycket på internet – av varierande kvalitet och aktualitet

## Webben

- <http://www.ida.liu.se/~TDDC76/>
- Meddelanden
- Allt kursmaterial
- Föreläsningsdagbok med föreläsningsplan
- Schema med planering och deadline
- m.m.

## Påminnelser

- Skarp anmälan till laborationerna under de första veckorna via WebReg
- Registrering på kursen?

## Översikt Datastrukturer och algoritmer

- Lista, stack, kö
- Träd: uttrycksträd (laboration 3), binära sökträd (enkla, AVL, Splay), B-träd
- Hashtabeller
- Prioritetsköer (binär heap)
- Sökning
- Sortering

## Objektorienterad programutveckling – miniprojekt

- Objektorienterad programmering
  - objekt och klass
  - härledning/ arv, polymorfi, dynamisk typkontroll, dynamisk typomvandling, ...
- Klass/objekt-relationer
  - association ("känner till")
  - sammansättning ("har")
  - arv ("är")
- Objektorienterad analys (OOA)
- Objektorienterad design (OOD)
- Dokumentationshjälpmedel
  - UML, t.ex. användningsfall, klassdiagram, ...
  - CRC-kort (klasskort)

## C++

- Bred användning – kraftfullt – intressanta konstruktioner – krävande – *mycket intressant!*

## Föreläsning 1-2

”Klassisk programmering”

Grunderna för kodning av algoritmer i många imperativa och objektorienterade språk:

- Funktioner – för modularisering
- Satser – för programstyrning på satsnivå
  - sekvens
  - val
  - repetition
- Uttryck – för enklare beräkningar
  - aritmetiska
  - jämförelse
  - logiska
- Variabler, konstanter – för datalagring
  - deklaration, räckvidd, synlighet, initiering, tilldelning, ...
- Klassiska datastrukturer
  - fältliknande (vector, array)
  - posttyper (**class**, **struct**)
  - länkad lista (poster + pekare + dynamisk minnestilldelning)

Detta är i princip förkunskaper tillsammans med att kunna lösa problem och formulera programmerbara lösningar för problem – görs i grundkurser.

Föreläsning 1-2 och Laboration 1 är i princip repetition av förkunskaper samt att lära sig grunderna i C++ – mer eller mindre nytt beroende på vad man programmerat i tidigare.

*Anm.* Laboration 1.5 *Frivillig uppgift* behöver även stoff från Fö 3 (klasser).