

Varmt välkomna till
Algoritmer och Datastrukturer!



Vad kursen innebär

- Vi kommer att titta på hur algoritmer fungerar, vilka klassiska algoritmer som finns för sökning och sortering och liknande, samt lära oss om data-strukturer och djupare koncept såsom rekursion
- Vi kommer även att fokusera lite mer på minne och minneshantering i datorer än kursen gjort tidigare
- Kursen är en fortsättningskurs som kommer att hjälpa er utvecklas som programmerare, och den är viktig för kommande kurser i t.ex .NET-utveckling och Arkitektur & Designmönster
- Vi förutsätter att ni har koll på grunderna i programmering och är bekanta med objektorientering vid det här laget: känner ni att ni är skakiga på den biten så ägna den här veckan åt att repetera så mycket ni kan

Lärare

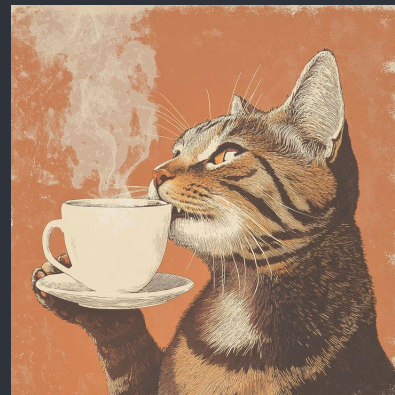
Carl-Johan Johansson
(kursansvarig)



Lovisa Johansson
(lärare)



Sara Herlitz
(amanuens)



Tidigare år

Från kursrapport VT25

Genomströmning VT25:

Antal studenter: 35 st (varav 4 var omregistrerade från föregående terminer)

Genomströmning 25 dagar efter kursens slut, V25: 48%

Genomströmning tidigare tillfälle:

V24: 58 %

V23: 44 %

V22: 57 %

V21: 48 %

V20: 62 %

Kommentar till genomströmning:

Genomströmning är lägre än snittet av de senaste åren. 56% av studenterna som skrev tentan blev godkända vilket är i nivå med snittet av de senaste åren men betydligt lägre än förra året när 83% blev godkända. Sedan VT24 ger tentan fler poäng medan laborationerna ger färre poäng.

Generellt sett en av de tyngre kurserna på programmet där man behöver arbeta mest per högskolepoäng: viktigt att vara på plats och följa med hela vägen.

Genomströmningen ligger kring 50% sett till tidigare år.

Det finns tydliga tecken på att generativ AI har haft en försämrande effekt på studenters inläring: de producerar bättre kod på laborationer men har färre egna kunskaper.

Problem: låg kodförståelse

Förra året hade många studenter problem med att fylla i en väldigt grundläggande metod på tentan, vilket innebar att de även missade nästan alla andra uppgifter som var kopplade till programmering:

```
/**
 * Find the smallest value in an list. If list n
 * contains { 3, 1, 0, 6 } the method will return
 * 0 since it is the smallest value in the list.
 *
 * @param l the list we want to search
 * @return the smallest value in the list
 */
public int findSmallest(LinkedList<Integer> l)
```

Lösningsförslag

```
/**
 * Find the smallest value in an list.
 * If list n contains { 3, 1, 0, 6 } the
 * method will return 0 since it is the
 * smallest value in the list.
 * @param l the list we want to search
 * @return the smallest value in the list
 */
public int findSmallest(LinkedList<Integer> l)
```

```
public int findSmallest(LinkedList<Integer> l)
{
    return Collections.min(l);
}

public int findSmallest(LinkedList<Integer> l)
{
    int smallest = Integer.MAX_VALUE;
    for (int number : l)
    {
        if (number < smallest) { smallest = number; }
    }

    return smallest;
}
```

Hur man klarar algokursen

E-tentasystemet Inspera låter oss skriva och kompilera kod från och med HT 25, och en stor del av tentan i algon kommer därför gå ut på att skriva fungerande kod som är körbar.

Det här gör det enklare för oss att testa kodförståelsen mer utförligt.



Se till att ni löser laborationer och uppgifter ärligt och självständigt under kursen, för det kommer att förbereda er inför examinationen.

Vi kommer även att ha mer gemensamma övningar under föreläsningarna än tidigare år för att förbereda er bättre för detta, men det enda sättet att bli bra på att koda är just att skriva mycket kod. **Lägg era utsatta 20 timmar i veckan på kursen. Kom på alla föreläsningar och labbpass. Ställ frågor. Be om hjälp om ni tycker att något är svårt: min dörr (och inkorg) är alltid öppen.**

Det var generellt väldigt dålig närvaro under labbtillfällena på Javan i höstas. Det finns en väldigt tydlig trend från alla tidigare kurstillfällen: **de studenter som klarar algon är de som dyker upp på föreläsningar och labbar.**

Vilseledning vid examination

Vilseledande vid examination (fusk) anmäls till disciplinnämnden och kan leda till varning eller avstängning upp till sex månader.

Att lämna in kod som man inte skrivit själv bedöms som fusk, oavsett om det är generativ AI eller en annan person som hjälpt till att skriva den.

Att dela med sig av sin lösning och att använda någon annans lösning bedöms båda som fusk.

Därför ska du aldrig diskutera eller dela kod från laborationsuppgifter med andra än din eventuella laborationspartner. Tänk också på att hålla dina Git-repositories privata om de innehåller labblösningar eftersom allting som är publikt på github också är tillgängligt för andra att plocka hem.

Vi hade samma regler redan på javakursen, och en del studenter dök ändå upp med AI-genererade labbar. Vi ser tydligt vad som är AI-genererat och vad som inte är det, och vi kommer vara hårdare med detta i algokursen.

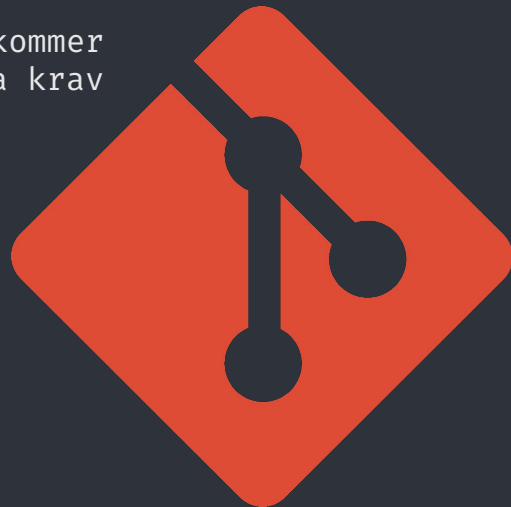


Versionshantering av kod

Alla laborationer förutsätter att ni använt versionshantering med regelbundna commits för att bygga upp er kod. För de som är lite osäkra på hur Git fungerar finns det en mängd resurser under modulerna: läs dessa, och ställ frågor om ni fortfarande är osäkra. IntelliJ har Git-integration som gör att ni slipper pusha kod via terminalen eller en separat GUI-klient.

Laborationer som inte har en tydlig versionshistorik kommer inte att godkännas. Studenter som inte uppfyller detta krav kommer att få göra en ny laboration.

Om tillfälle finns kommer vi även att kika lite närmre på lite mer avancerade Git-funktioner under kursens gång, såsom rollbacks och “forking” (att arbeta på separata förgreningar).



IDE och miljöer

Vi använder IntelliJ på kursen (och programmet rent generellt). Ni är fria att använda andra IDE om ni vill, men IntelliJ är det enda som vi kommer erbjuda hjälp med. Det är också med råge det bästa IDE:t för Javaprogrammering, så vi rekommenderar att ni bekantar er med det.

Även om ni väljer att använda ett annat IDE kommer ni behöva IntelliJ under kursens senare veckor när vi börjar arbeta med testning, så se till att ni har programmet installerat.

Installation av JDK (Java Development Kit) sker automatiskt via IntelliJ och beskrivs i guiden under modulerna.

Vill ni använda en annan editor för kod behöver ni själva installera ett JDK och lära er hur ni lägger till det i era systemvariabler.

Ni som precis gått Inledande Programmering Med Java har redan allt ni behöver installerat på era datorer och behöver inte göra några ändringar.

Projektet, vecka 11

Genomförs individuellt eller i par (max två personer).

Du opponerar individuellt på en annan inlämning. Ordinarie opponeringstillfälle sker på sal. Uppsamlingstillfällena sker via Zoom.

Om din inlämning och/eller opposition innehåller mindre problem har du möjlighet att komplettera skriftligt. Om problemen är större behöver du åtgärda problemen och sedan göra en nya inlämning och opposition på ett kommande uppsamlingstillfälle.

Två uppsamlingstillfällen erbjuds för projektet, ett under våren och ett innan höstterminen startar. Om du vill lämna in projektet efter detta datum behöver du vänta till nästa kurstillfälle.

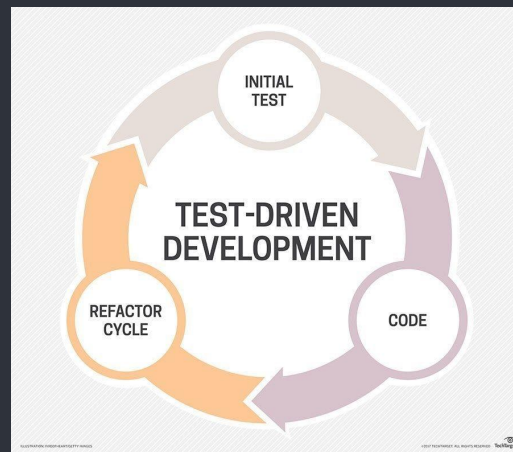
Ändring från i fjol: Test-Driven Development

I år kommer vi att kika litegrann på TDD på kursen: detta är ett nytt moment och ingenting som är examinerande eftersom det inte finns med i kursmålen, men det är någonting som jag själv ville inkludera.

Tidigare kursstudenter har även uttryckt att de önskade att de fått en bättre introduktion till testning, hur man skriver test, och hur man kan använda dem för att underlätta utveckling av program.

Vi kommer att köra en vecka TDD mot slutet av kursen.

Ni behöver inte läsa något eller förbereda er inför den här veckan: Det räcker med att ni dyker upp så visar jag hur ni sätter upp en testmiljö, och sen kommer vi göra lite gemensamma övningar i hur man kan använda TDD för att skapa och förbättra en algoritm inkrementellt.



Tentamen

E-tentamen i sal (4 h).

Tentamen innehåller frågor som liknar övningsuppgifterna. Vi rekommenderar därför att du alltid gör dessa uppgifter eftersom de förbereder dig inför både tentan och laborationerna.

Två omtentamen erbjuds, en under våren och en innan höstterminen startar. Ytterligare tillfällen erbjuds under nästa kurstillfälle.

Det går att skriva tentan på annan ort (annat lärosäte eller lärcentra). De flesta lärcentra är betydligt mer flexibla och tillmötesgående än lärosäten.

När du säkerställt att du kan skriva tentan på annan ort hör du av dig till kursansvarig med kontaktuppgifter till aktuellt lärcentra/lärosäte.

Om man har Javalabbar kvar

Eftersom jag kommer vara själv på stora delar av kursen kommer jag behöva prioritera algon, och jag kommer därför inte hinna ta några javaredovisningar under kursens aktiva period. Lovisa är också upptagen med Databaser 2 och kommer inte hinna med något annat.

Har man javalabbar kvar från i höstas och vill redovisa dem kan man i stället kontakta Sara, så får ni redovisningstid av henne så fort hon hinner: sara.herlitz@im.uu.se

Kan ta någon vecka beroende på hur mycket hon har att göra, men det går mycket fortare än att behöva vänta till uppsamlingstillfället i början av Juni. Vi rapporterar in poängen så fort ni blivit godkända så att ni har dem inför kommande kurser.

För er som läser kursen på distans

Varmt välkomna! Skicka gärna ett mail till mig och introducera er lite kort så att jag har koll på vilka ni är!

Eftersom ni tyvärr inte kan vara med under föreläsningar och labbpass behöver ni i högre grad arbeta självständigt med kursen. Föreläsningsslides och kod-exempel kommer att publiceras i Studium och jag har gjort mitt bästa för att dessa ska vara så utförliga som möjligt.

Det är dock omöjligt att täcka in allt vi går genom muntligt på presentationen. Om det är någonting som är oklart eller om det är information som ni känner saknas så får ni gärna höra av er.

Behöver ni mer information om att skriva tentamen på annan ort så hör av er till Studieadministrationen:

När ni vill redovisa labbar får ni gärna höra av er till antingen Carl-Johan, Lovisa eller Sara så bokar vi redovisningstillfällen via Zoom.

För frågor om handledning eller föreläsningar vore det dock bättre om ni kontaktar Carl-Johan eftersom han ansvarar för lektionsupplägget och vet vad vi specifikt har gått genom.