

Läroplan för gymnasieskolan

Lgy⁷⁰

GÖTEBORGS UNIVERSITETSBIBLIOTEK ✓



100172 4692

Datakunskap

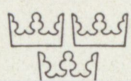
DESSA BOK TILLHÖR RETRACKSBOKLIOTEKET
PÅ GÖTEBORGS UNIVERSITET

II Supplement 95

SKOLÖVERSTYRELSEN 1983

Föreliggande supplement i alternativämnet Datakunskap på treårig naturvetenskaplig linje skall tillämpas fr o m läsåret 1983/84.

BIBLIOTEK GÖTEBORGS UNIVERSITET



Pedagogiska biblioteket

Supplement 95

GÖTEBORGS UNIVERSITETSBIBLIOTEK



14000

000497916

Lgyll⁷⁰

Läroplan för gymnasieskolan

SKOLÖVERSTYRELSEN

Liber Utbildningsförlaget Stockholm

Supplement 95

Fastställt 1983-09-16

Dnr 5050-83:2014

Datakunskap

Liber Utbildningsförlaget
162 89 STOCKHOLM

Separata exemplar kan beställas genom
Liber distribution
Order Utbildning
162 89 STOCKHOLM
Tfn 08 - 739 91 00

FÖRORD

Läroplanen för gymnasieskolan (Lgy 70) består av en allmän del (del I), som är gemensam för samtliga studievägar, samt av supplement (del II) för skilda studievägar och ämnen.

Den allmänna delen (del I) innehåller av Kungl Maj:t fastställda mål och riktlinjer, timplaner och kursplaner (mål och huvudmoment i enskilda ämnen) samt av SÖ utfärdade allmänna anvisningar för gymnasieskolans verksamhet.

Supplementdelen (del II) återger timplaner och kursplaner (mål och huvudmoment), fogar till dessa i förekommande fall delmoment och årskursfördelningar samt ger allmänna riktlinjer för undervisningens bedrivande i de olika ämnena.

Föreliggande supplement i alternativämnet Datakunskap på treårig naturvetenskaplig linje skall tillämpas fr o m läsåret 1983/84.

Med tanke på den fortlöpande läroplansöversynen är det angeläget att erfarenheter av läroplanens tillämpning som görs på skolorna delges SÖ.

Stockholm i november 1983

Skolöverstyrelsen

© 1983 Skolöverstyrelsen och
Liber Utbildningsförlaget

ISBN 91-40-71108-0 1 2 3 4 5 6 7 8 9 10

Liber Tryck Stockholm 1983 304154

INNEHÅLL

MÅL 6

HUVUDMOMENT 6

ALLMÄNNA KOMMENTARER 6

KOMMENTARER TILL INNEHÅLLET, servicematerial 9

1. Datorsystem 11

Tal- och teckenrepresentation 11

Datorns uppbyggnad och funktion 11

Kringutrustning 12

Systemprogramvara 12

Programspråksnivåer 12

2. Programutvecklingsmetodik och programmering 13

Programmering 13, 15

Programutveckling 14, 16

Dokumentation 14, 16

3. Datoranvändning i naturvetenskap 19

Mätning och styrning 19

Beräkning och bearbetning 19

4. Samhällsaspekter på datoranvändning 20

Arbetsmarknaden 20

Integritet och sårbarhet 20

Informationssamhället 20

Aktuell debatt 21

5. Systemering 22

Introduktion och grundbegrepp 23

Systemeringsprojekt 23, 25

MÅL

Genom undervisningen i datakunskap skall eleven

- förvärva grundläggande kunskaper om maskinvaran och programvaran i ett datorsystem,
- bli väl förtrogen med hur datorer kan användas,
- förvärva kunskaper i problemanalys och programmering,
- förvärva färdighet i att använda datorer och data-logiska metoder i olika tillämpningar,
- förvärva färdighet i att kommunicera i ADB-frågor,
- få erfarenhet av arbete med ADB-projekt i arbetslag,
- förvärva förmåga att realistiskt bedöma datateknikens möjligheter och begränsningar,
- förvärva en helhetssyn på datoriseringens konsekvenser med hänsyn till såväl individ som samhälle.

Huvudmoment

Datorsystem
 Programutvecklingsmetodik och programmering
 Datoranvändning i naturvetenskap
 Samhällsaspekter på datoranvändning
 Systemering

Allmänna kommentarer

Studiernas syfte

Flertalet elever kommer i sin framtida yrkesverksamhet eller som samhällsmedborgare att möta ADB (automatisk databehandling) i en eller annan form. De kan behöva använda ADB som ett hjälpmedel inom det egna yrket eller enbart fungera som användare av resultat från ADB (listor, dokument). De kan också komma i en situation där de aktivt måste medverka vid införandet av ett ADB-system. Undervisningen i datakunskap bör därför utformas så, att eleverna förvärvar datamognad som användare. Avsikten är alltså inte att utbilda programmerare.

Arbetsätt

Undervisningen i datakunskap skall vara färdighetsinriktad och kännetecknas av aktivt elevarbete med övningsuppgifter. Nya begrepp, metoder och tekniker inhämtas genom praktisk övning. Endast i vissa grundläggande avsnitt kan helt lärarledd undervisning vara att föredra.

Man bör sträva efter att successivt öka omfånget på övningsuppgifterna från enkla övningar över större tillämpningsuppgifter till projekt. Med projekt avses ett målinriktat uppdrag, begränsat till tid och resurser. Projektuppgiften bör ha en sådan omfattning, att eleverna måste presentera en plan för projektets genomförande innan själva arbetet startar samt eventuellt en eller flera etappredövningar under arbetets gång. Projekt bör i allmänhet genomföras som grupparbete.

Planering och samverkan

Huvudmomenten är inte ordnade i kronologisk ordning utan bör integreras på ett naturligt sätt. Speciellt bör observeras att momentet samhällsaspekter tas upp fortlöpande under kursen och då anknyts till övriga moment.

Ämnet datakunskap bör också vara ett hjälpmedel vid studiet av andra ämnen, varför man bör sträva efter att hämta tillämpningsuppgifter från dessa.

Genom projekt, där datalogiska metoder används för att lösa en uppgift, får man en naturlig samverkan mellan olika ämnen och olika elevgrupper.

I många sammanhang kan det vara lämpligt att använda texter på engelska.

Anknytning till omvärlden

Det är viktigt att eleverna får en så korrekt uppfattning som möjligt om ADB-teknikens nuvarande och framtida roll i samhället.

Större uppgifter och projekt bör anknytas till verkliga situationer.

Förberedda studiebesök på företag eller institutioner med ADB-verksamhet bör förekomma. Det är då värdefullt om eleverna ges tillfälle att intervjua ledning och personal.

Samtliga elever bör om möjligt någon gång under tiden från vårterminen i årskurs 2 till och med höstterminen i årskurs 3 beredas en sammanhängande praktikperiod med ADB-verksamhet om minst 2 veckor.

Momentbeskrivning

Efter varje rubrik anges inom parentes förslag till antal undervisningstimmar för avsnittet.

Kommentarer till innehållet

1. DATORSYSTEM

Huvudvikten läggs vid förståelse och funktion, inte vid tekniska aspekter. De olika delmomenten konkretiseras, där så är möjligt, med hjälp av skolans datorsystem. Delmomenten behöver ej behandlas i ett sammanhang.

<u>Innehåll</u>	<u>Kommentarer och exempel</u>
<u>1.1 Tal- och teckenrepresentation</u>	
Olika talsystem	Med utgångspunkt från decimalsystemet behandlas uppbyggnaden av andra talsystem, främst det binära och hexadecimala.
Bit, bitgrupp (byte) och ord	Olika möjligheter att i ett datorsystem representera två symboler exemplifieras. Begreppet ordlängd behandlas.
Representation av heltal och flyttal	Exempelvis visas en binär representation av ett heltal och ett flyttal med tecken i ett 32-bitarsord. Talområde och noggrannhet diskuteras.
Kodsystem	De vanligaste koderna för representation av tecken (bokstäver, siffror och andra symboler) beskrivs.
<u>1.2 Datorns uppbyggnad och funktion</u>	
Historisk utveckling	Kortfattad historik från ENIAC till dagens datorsystem.
Centralenhet (CPU)	I samband med genomgången av centralenheten (CPU) behandlas olika typer av register.
Primärminne	Principerna för läsminne (ROM) och skriv- och läsminne (RWM) bör behandlas. Man bör också nämna något om PROM och EPROM.
Datatransport och datalagring	In-utkretsarnas uppgift beskrivs. Serie- och parallellöverföring kan diskuteras. Med hjälp av en figur visas hur instruktioner och data transporteras mellan primärminne och register när ett programsteg exekveras. Klockpulsernas funktion nämns.
Olika typer av datorer och datorsystem	Generella datorer i motsats till specialdatorer diskuteras, lämpligen i anslutning till studiebesök. Även datornät kan behandlas.

InnehållKommentarer och exempel1.3 Kringutrustning

Sekundärminnen

Behandlingen av sekundärminnen kan inledas med en kort översikt över databärare. För- och nackdelar hos olika typer av sekundärminnen diskuteras.

Skrivare, dataskärmar och tangentbord

Vid genomgången av kringutrustning läggs huvudvikten vid den typ av utrustning som ingår i skolans datorsystem.

1.4 Systemprogramvara

Operativsystem och hjälpprogram

Vikten läggs vid operativsystemets uppgift att göra maskinen brukbar för användaren och att fördela datorsystemets resurser på bästa sätt. I samband härmed behandlas olika typer av driftsformer (realtidsbearbetning etc). Länkare, laddare och andra hjälpprogram kan demonstreras om de ingår i skolans datorsystem.

Översättare och redigerare (editorer)

Interpretering och kompilering behandlas. Begreppen källprogram och objektprogram kan tas upp.

Filhanteringssystem och databassystem

Filhanterings- och databassystem demonstreras i den omfattning de ingår i skolans datorsystem.

1.5 Programspråksnivåer

Maskin- och assembleringspråk

Principiella skillnader mellan lågnivå- och högnivåspråk tas upp.

Högnivåspråk

En kort översikt över olika högnivåspråk bör göras, speciellt med tanke på deras användningsområden.

2. PROGRAMUTVECKLINGSMETODIK OCH PROGRAMMERING

Metodisk programutveckling är ett relativt nytt kunskapsområde under stark utveckling. Ämnet ingår ej i någon lärarutbildning för gymnasieskolan. Av denna anledning har kommentarerna till detta moment gjorts fylliga, så att de skall ge oerfarna lärare hjälp med översikt över ämnet och stöd för en metodisk planering av undervisningen.

Programutvecklingsmetodik är kunskapen om hur programvara framställs på ett systematiskt sätt, från problemanalys och metodval till färdigkodat och testat program med dokumentation (programvara). En arbetsplan för detta återfinns på sidan 13.

Ur metodisk synpunkt kan kursen indelas i tre nivåer: grundbegrepp, fördjupning och projekt. På varje nivå integreras delmomenten programmering, programutveckling och dokumentation. Jämsides med inläringen av de olika begreppen i ett programspråk görs eleverna förtrogna med och vänjs vid att tillämpa de olika momenten i den ovan nämnda arbetsplanen för programutveckling. Inläringen underlättas i hög grad av ett programspråk med möjlighet till underprogram (se sid 10) och lämpat för strukturerad programmering. För att kursen ej skall bli för arbetskrävande övas endast några moment under varje programmeringsuppgift. Fullständig redovisning krävs ej för alla uppgifter.

Gränsen mellan en omfattande övningsuppgift och ett projekt är flytande. Därför går det ej heller att dra någon skarp gräns mellan nivåerna 2 och 3, utan eleverna vänjs successivt vid uppgifter av allt större omfattning. Varje elev skall gå igenom alla tre nivåerna.

NIVA 1. GRUNDBEGREPP

<u>Innehåll</u>	<u>Kommentarer och exempel</u>
2.1 Programmering	
Program med strängar och strängvariabler: - utskrift - tilldelning - inmatning - enkel stränghantering - relationsoperatorer - selektion (val) - iteration (upprepning) med explicit villkor	De vanligaste instruktionerna och de enklaste formerna för datarepresentation lärs in genom enkla programmeringsövningar. Samtidigt får eleverna lära sig att använda skolans datorsystem. Under förutsättning att det programspråk som används har instruktioner för direkt, enkel stränghantering bör de första programmeringsövningarna göras med strängar och strängvariabler. Om så ej är fallet kan det vara bättre att först arbeta med program med tal och numeriska variabler.
Program med tal och numeriska variabler: - algebraiska operatorer - relationsoperatorer - hopp - räkningslingor (iteration med implicit villkor)	I exemplen med utskrift bör enkel styrning, såsom hopskrivning av text, standardtabulering och radmatning ingå. Inmatning bör omfatta såväl interaktiv inmatning som läsning från datasatser. Stränghanteringen bör bli omfatta sammansättning, delsträng och längd av sträng.

InnehållKommentarer och exempel

övningarna kan varieras mellan

- läsning och förklaring av färdiga program,
- "skrivbordsexekvering" av program, egna och andras,
- komplettering/rättning av ofullständiga/felaktiga program,
- framställning av egna program.

De egna programmen består till en början av endast ett par satser med tonvikten lagd på en enda ny instruktion. De sista uppgifterna bör resultera i program som innehåller de flesta av de instruktioner som ingår i nivå 1.

2.2 Programutveckling

Strukturerad programmering

Det är viktigt att man redan från början betonar god programstruktur.

De logiska basstrukturerna:

- sekvens
- selektion
- iteration

Vid programmeringsövningarna uppmärksammas möjligheterna att koda basstrukturerna med hjälp av det använda språkets instruktioner.

Problemdefinition

Under nivå 1 bör problemdefinition enligt "arbetsgång vid utveckling av programvara" (se sidan 13) börja övas.

2.3 Dokumentation

Dokumentation i programlistan

I de första övningarna, med mycket enkla program, bör dokumentationen endast bestå av en programlista, försedd med identifieringsinformation (programkonstruktör, programversion, datum), variabelbeskrivning samt eventuella övriga kommentarer.

Minimidokumentation:

- problemdefinition
- programplan
- programlista (med identifieringsinformation och kommentarer)
- variabellista

Senare bör eleverna vänjas vid att alltid redovisa enligt vidstående krav för minimidokumentation. Problemdefinitionen kan då redovisas enligt "arbetsgång vid utveckling av programvara" och programplanen redovisas i grafisk form eller som pseudokod. Variabellisten kan eventuellt redovisas i programlistan.

Mot slutet av nivå 1 bör eleverna vara vana vid att

- programplanen alltid redovisas innan kodningen påbörjas,
- varje programmeringsuppgift slutredovisas enligt angivna minimikrav.

NIVA 2. FÖRDJUPNING

Kunskaperna i det aktuella programspråket fördjupas genom tillämpningsuppgifter av sådan omfattning att de med fördel organiseras med underprogram(*). Samtidigt behandlas några grundläggande datatekniska bearbetningsmetoder. Varje övningsuppgift bör vara riktad, d v s konstruerad för att belysa någon eller några bestämda bearbetnings- och/eller datalagringsmetoder.

Vid val av övningsexempel bör man sträva efter att hämta innehåll och uppgifter från andra ämnen och ämnesområden. Därigenom uppmärksammas eleverna (och företrädarna för andra ämnen) på de möjligheter till ämnessamverkan som ligger i, att man inom datatekniken tar fram metoder och system för insamling, bearbetning och presentation av data för användning i andra ämnen.

Fullständigt genomförd programutveckling är tidskrävande. För att eleverna skall hinna arbeta med hela det angivna kursinnehållet kan uppgifterna varieras mellan

- rena tillämpningar för studium av färdiga programpaket,
- tillverkning av någon tillämpningsmodul till ett i övrigt färdigt programpaket,
- urval och sammansättning av ett antal "standardmoduler" (grundprogram) till ett eget programpaket för någon speciell tillämpning,
- fullständig utveckling av egna program.

Innehåll

Kommentarer och exempel

2.1 Programmering

Instruktioner och datastrukturer:

- flervälsinstruktioner
- listor, tabeller
- avancerad stränghantering
- logiska variabler
- underprogram
- filer

I stränghantering bör bl a ingå konvertering mellan tecken och ASCII-kod samt mellan strängvariabel och numerisk variabel, delsträng samt sökning i sträng.

Eventuellt kan logiska variabler behandlas.

Underprogram(*) införs tidigt. Eleverna vänjs vid att vid varje programmeringsuppgift överväga hur de skall användas för att ge programmet bästa tänkbara struktur.

För att möjliggöra mer varierade uppgifter bör filbegreppet och användning av sekventiella filer införas tidigt. Arbete med filer kräver att man från varje arbetsplats har tillgång till skrivminne. Även direktaccessfiler kan behandlas.

Bearbetningsmetoder:

- beräkning
- sökning
- sortering

Beräkningstillämpningar anknyts till kursen i matematik och till moment 3.

De enklaste metoderna för sökning (linjär och binär) och sortering ("jämför grannar och byt") behandlas i samband med listor och tabeller. Även någon avancerad sorteringsmetod kan tas upp.

*) Med underprogram avses procedurer/subrutiner samt funktioner. Underprogrammet utgörs av en sekvens satser som (i de flesta programspråk) ges ett eget namn att anropas med från huvudprogrammet. I allmänhet kan parametervärden överföras mellan huvudprogram och underprogram.

InnehållKommentarer och exempel

Användarvänlighet och presentationsmetoder:

- felhantering
- menyteknik
- grafik

Eleverna uppmärksammas på vikten av att programmen är operatörsvänliga (bildskärmlayout, naturlighet i inmatning) och har en väl genomförd felhantering.

2.2 Programutveckling

Fullständigt genomförd programutveckling

De olika stegen i "arbetsgång vid utveckling av programvara" övas i anslutning till olika programmeringsuppgifter. Av tidsskäl övas endast något steg i programutvecklingsmetodikerna på varje uppgift. Fullständigt genomförd programutveckling görs på en eller ett par uppgifter mot slutet av nivå 2.

2.3 Dokumentation

Framställning av fullständig dokumentation (se punkt 6, sid 13)

Framställning av olika delar av programdokumentation övas i anslutning till olika programmeringsuppgifter. Dokumentation får aldrig bli ett självändamål. Av tidsskäl framställs därför endast någon del av dokumentationen på varje uppgift. Fullständig dokumentation görs dock på en eller ett par uppgifter mot slutet av nivå 2. Eleverna kan få inse vikten av god dokumentation om de får i uppgift att läsa och förklara andras program.

NIVA 3. PROJEKT

Nedanstående kommentarer gäller rena programmeringsprojekt. I de fall man arbetar med fullständiga projekt (systemering och programmering) gäller kommentarerna programmeringsdelen (från programförutsättningar till färdig programvara). Kommentarer till systemeringsdelen av ett fullständigt projekt återfinns under moment 5, nivå 3.

Syftet med projektarbetet är

- att eleverna skall få tillämpa och fördjupa kunskaperna i programstrukturering, programmering och dokumentation,
- att konkret visa på behovet av ett disciplinerat arbetssätt (problemstrukturering, väldefinierad arbetsfördelning, strikt dokumentation etc) vid programproduktion i arbetslag,
- att tillgodose elevernas olika intresseinriktningar genom att ge valmöjlighet mot olika tillämpningsområden,
- att ge möjlighet till ämnessamverkan.

I projektarbetet kan eleverna ges tillfälle

- att ytterligare studera olika programvarukrav, såsom felkontroll och felhantering, testkörning, användarvänlighet (för såväl operatör som systemansvarig) etc,
- att använda färdiga programmoduler för t ex beräkning, statistik, grafik, sortering, filhantering etc,
- att använda och studera färdiga programpaket och -system av tillämpningskaraktär, såsom beräknings- och statistikprogram, sorterings- och filhanteringsprogram, ordbehandlings- och databassystem etc.

Vid val av projektuppgifter bör man eftersträva samverkan med andra ämnen och ämnesområden samt söka tillgodose elevernas individuella intressen. Ett sätt att väcka intresse för samverkan hos företrädare för andra ämnen och samtidigt stimulera eleverna och ge dem tillfälle att fundera över olika uppgifter är att, som en inledning till kursavsnittet, låta eleverna göra en inventering av tänkbara uppgifter och sammanställa dem i en projektkatalog.

Det är vanligt att programmeringsprojekt blir mer omfattande än man från början inser. Därför bör läraren se till, att underlaget är väl definierat (så att inte en stor del av arbetet blir systemering istället för programmering) samt att uppdraget som helhet begränsas. Detta bör uppmärksammas redan vid val och fördelning av uppgifterna, men även fortlöpande under arbetets gång, t ex genom projektplans- och etappredovisningar (se nedan).

Ett projekt bör alltid inledas med att arbetsgruppen gör en projektplan med etappindelning. Först sedan planen godkänts påbörjas projektarbetet och genomförs med ett antal etappredovisningar före slutredovisning. Krav på strukturering och dokumentation enligt tidigare övade principer bör ställas.

Innehåll

Programmeringsprojekt i arbetslag inom ett eller flera av områdena

- administrativ databehandling,
- teknisk databehandling,
- mätning och styrning

Kommentarer och exempel

Varje elev bör genomföra minst två projekt i olika arbetslag. Ett av projekten bör genomföras i ett stort lag (5-8) för att ge erfarenhet av de speciella krav på planering (väl definierade arbetsuppgifter för varje deltagare samt väl definierade "gränssnitt" mellan de olika uppgifterna) som dylikt arbete ställer.

Exempel på projektuppgifter:

- Administrativ databehandling: Skoladministrativa rutiner (elevregister, läroboksregister, betygsstatistik) för skolans expedition, övnings- och testprogram för glosor och grammatik i språk, Start- och resultatlistor för friluftsdagar i idrott.
- Teknisk databehandling: Beräkningar, statistik och grafik i samverkan med matematik, naturvetenskapliga, tekniska och ekonomiska ämnen samt samhällskunskap.
- Mätning och styrning: Mät- och reglertillämpningar i försöksuppställningar i samverkan med naturvetenskapliga och tekniska ämnen.

Ett förslag till arbetsgång vid utveckling av programvara

1 Problemdefinition

- Vad skall programmet ge för resultat (Utdata:Innehåll,form)?
- Vilka är förutsättningarna (Indata,begränsningar i systemet)?
- Vad skall programmet göra (Dataomformning)?

2 Metodval

- Programmets funktion i några få steg i klartext.
- Dataflödet i några få steg i klartext.

3 Stegvis detaljering (enligt top-down-metod)

- * Programplan (slutprodukt).
 Detaljerad och överskådlig programplan, grafisk eller i pseudokod, med kommentarer.
 Kravet på överskådlighet innebär att planen är indelad i
 - underprogram (procedurer/subrutiner och funktioner) och
 - moduler (överblickbara programavsnitt med en bestämd uppgift).
 Alla programdelar bör ha en ingång och en utgång samt väldefinierad dataöverföring mellan delarna.
- * Dataplan (slutprodukt).
 Detaljerad och överskådlig
 - beskrivning av använda datastrukturer (i primärminne och på sekundärminne) samt
 - grafisk dataflödesplan (över datamängderna och deras behandling).

4 Kodning i högnivåspråk

överskådlighet enligt punkt 3 i väl avgränsade delar med kommentarer och hänvisning till dokumentationen (punkt 6).

5 Utprovning

- Utprovning kan göras stegvis på delar av programmet med "dummys" för ännu ej kodade programdelar och för data från dessa.
- Bearbetning med testdata görs på hela programmet.

6 Dokumentation

- Problemöversikt
- Beskrivning av använda metoder
- Användarhandledning med bearbetningsexempel
- Programbeskrivning med programplan och programlista
- Beskrivning av datastrukturer och dataflöde
- Handledning för systemansvarig

 Anmärkning.

Stegvis detaljering enligt top-down-metod är den allmänt accepterade och mest använda metoden vid utveckling av program av "någorlunda" omfattning. I vissa fall kan någon annan metod vara tillämplig, t ex bottom-up, när man från början vet att en viss rutin kommer att behövas på flera platser i programmet, eller när man vill utgå från en bestämd rutin och bygga upp sitt program omkring denna. Bottom-up-metoden bör dock endast användas vid framställning av mindre program eller delar av program. Så snart det är fråga om stora program bör top-down-metodiken prägla programmeringsarbetet så, att man i största möjliga utsträckning arbetar från översikt mot detaljering.

3 DATORANVÄNDNING I NATURVETENSKAP

Innehåll	Kommentarer och exempel
3.1 Mätning och styrning	
Olika typer av givare	Det är lämpligt att demonstrera de givare eleverna senare skall laborera med, t ex termistorer, fototransistorer.
Anslutning av mät- och styrutrustning	Någon metod för hur givare ansluts till en dator med hjälp av gränssnitt (interface) demonstreras. Gränssnittets uppgift att anpassa och organisera signalerna mellan givare och dator bör nämnas.
Laborationer	<p>Eleverna bör göra några laborationer som dels visar grundläggande principer vid mätning och styrning, dels visar möjligheterna att mäta över korta och långa tidsintervall. En del av dessa laborationer görs lämpligen på fysik-, kemi- eller biologilinstitutionerna i samverkan med respektive ämnen.</p> <p>Några exempel:</p> <ul style="list-style-type: none"> - Undersökning av hur man med hjälp av program eller kommandon sänder ut och tar emot signaler. - Kalibrering av utrustning, uppkopplad för mätning av t ex spänning, temperatur eller tid. - Mätning och styrning av t ex temperatur.
Industriella tillämpningar	En kortfattad översikt över aktuella tillämpningar ges, gärna i samband med studiebesök.
3.2 Beräkning och bearbetning	
Felkällor vid beräkning med dator	Avrundningsfel, trunckeringsfel och kancellation behandlas.
Numeriska metoder	Egna och färdiga program används för exempelvis ekvationslösning, numerisk integration och numerisk lösning av differentialekvationer. Samverkan sker med matematik. Diskreta metoder (stegmetoder) kan tillämpas i fysik (t ex fritt fall med luftmotstånd) och biologi (t ex populations-tillväxt).
Simuleringar och statistiska bearbetningar	Problemlösning genom simulering övas i några enkla fall. Stora datamängder, t ex från en enkätundersökning, bearbetas. Principerna för linjär optimering, linjär anpassning och korrelation kan gås igenom kortfattat. I så fall bör några uppgifter lösas med hjälp av färdig programvara.

4. SAMHÄLLSPÅVERKningar PÅ DATORANVÄNDNING

Momentet innebär en fördjupning av elevernas tidigare kunskaper i datalära. Eleverna skall skaffa sig en överblick över hur datorer används idag och en beredskap inför tänkbara framtida användningar. För att uppnå en helhetsbild av samhällets datorisering kan eleverna, exempelvis i projektform, få studera den totala datoriseringen inom någon sektor av samhället. Momentet planeras i samverkan med ämnet samhällskunskap.

Innehåll

Kommentarer och exempel

4.1 Arbetsmarknaden

Datorns framväxt i samhället

Utvecklingen av datorer och robotar sedd mot utvecklingen i samhälle och näringsliv samt analys av drivkrafterna. Dataindustrin i Sverige och övriga världen.

Användning av datorer idag i arbetslivet

Datoriseringens inverkan på arbetsmiljö och arbetsinnehåll kan undersökas vid studiebesök i olika dator- och robotmiljöer. Dessa analyseras ur exempelvis sociala, ergonomiska och ekonomiska aspekter.

Sysselsättning

Datoriseringens inverkan på antalet arbetstillfällen och på arbetsplatsernas lokalisering (t ex till hemmen) diskuteras och analyseras.

Arbetsmarknadsparterna och medbestämmandelagen

Eleverna kan, t ex med hjälp av material utgivet av LO, TCO, SAF m fl organisationer, studera arbetsmarknadsparternas ställningstaganden i frågor som rör datoriseringens effekter. Möjligheten att påverka datoriseringen med hjälp av medbestämmandelagen (MBL) diskuteras.

4.2 Integritet och sårbarhet

Datalagen

Eleverna bör få grundlig kännedom om innehåll och syften hos datalagen och angränsande lagar samt om Datainspektionens uppgifter.

Samhällets sårbarhet

Samhällets sårbarhet, som en följd av datoriseringen och det beroende den skapar, diskuteras. Man kan behandla teknikberoendet vad gäller maskin- och programvara, utlandsberoendet beträffande datorer, komponenter och programvara etc.

ADB-säkerhet

Här behandlas fysisk säkerhet, funktionssäkerhet, datasäkerhet och kvalitetssäkerhet hos datasystem.

4.3 Informationssamhället

Historisk utveckling

Den tekniska utvecklingen, med ökade möjligheter att lagra och överföra stora datamängder, studeras. Man kan bl a diskutera "informationsexplosionen", "det papperslösa samhället" och internationella informationssystem.

Innehåll	Kommentarer och exempel
Teledatasystem	Eleverna informerar sig om teledatasystem i Sverige och i andra länder samt diskuterar systemens inverkan på tidnings- och bokutgivning, på sociala relationer i samhället och inom familjen, eventuella effekter på språket etc.
Artificiell intelligens	Inom området artificiell intelligens (AI) kan man t ex diskutera <ul style="list-style-type: none"> - vilka konsekvenser medför införandet av s k expertsystem för yrkeskunskap och kreativitet? - inom vilka områden bör datorer utnyttjas för att förstärka eller ersätta mänsklig aktivitet? - i vilken utsträckning bör man utnyttja datorsimulering i psykologisk forskning?
Datoriseringen och demokratin	Eleverna bör behandla statsmakternas, kommunernas, de politiska partiernas, fackorganisationernas, företagens, hushållens och individernas möjligheter att styra eller påverka datoriseringen och utvecklingen av informationsteknologin. Vidare kan t ex diskuteras om datormodeller som beslutsunderlag och "elektroniska folkomröstningar" är förenligt med demokratin.
4.4 <u>Aktuell debatt</u>	
Artiklar och rapporter	Eleverna ges tillfälle att med hjälp av artiklar i dagspress och fackpress, rapporter från institutioner och statliga verk etc, följa den aktuella debatten i datoriseringsfrågor, t ex om den framtida utvecklingen.
Referat och diskussioner	Enskilda elever eller elevgrupper kan ges i uppdrag att sammanställa och referera press-, radio- och TV-debatten i någon aktuell fråga, t ex som inledning till en diskussion.

9 SYSTEMERING

Systemering är ett relativt nytt kunskapsområde under stark utveckling. Någon allmänt accepterad metod existerar ej. Ämnet ingår ej i någon lärarutbildning för gymnasieskolan. Av denna anledning har kommentarerna till detta moment gjorts fylliga, så att de skall ge oerfarna lärare hjälp med översikt över ämnet och stöd för en metodisk planering av undervisningen.

Systemering (systemutveckling) är arbetet att konstruera och införa ett informationsbehandlingssystem, som till större eller mindre del kan vara datorbaserat. I denna kurs inriktas arbetet mot system för automatisk databehandling. Arbetsmetodikerna inom andra områden är dock likartad.

Det är viktigt att i momentet systemering även behandla frågor som gäller ergonomi, arbetsmiljö och medbestämmande.

Ur metodisk synpunkt kan kursen indelas i tre nivåer: introduktion, styrt projekt och fritt projekt.

Större delen av tiden ägnas åt grupparbete med systemeringsprojekt för att eleverna skall lära sig arbetsmetodik och tekniker praktiskt samt få övning i projektarbete i arbetslag.

Lärlarledda lektioner kan förekomma

- för genomgång av de grundbegrepp som behövs för att eleverna skall kunna börja arbeta praktiskt,
- för presentation av projektuppgifterna,
- under det styrda projektet för att sammanfatta och befästa använda metoder och tekniker, samt för att presentera nya,
- under det fria projektet i samband med eventuella gemensamma presentationer av projektetapper samt slutredovisningar.

Det bör betonas att den här beskrivna uppläggnngen av kursen är ett alternativ av många. Den naturliga formen för undervisning i systemering är dock projektarbetet, varför, oberoende av kursuppläggnngen, större delen av tiden bör ägnas åt detta. Man kan t ex ersätta det styrda projektet med något fler lärlarledda lektioner samt ett mer omfattande fritt projekt eller, om systemeringsuppgiften väljs lämpligt, utöka projektet till att omfatta även programmering av uppgiften (fullständigt projekt).

Även ifråga om val av systemeringsmetod kan uppläggnngen varieras mellan t ex enkel input-output-analys, någon etappmetod och experimentell systemutveckling ("systemskisser med dator").

NIVA 1. INTRODUCTION OCH GRUNDBEGREPP

Delmomentet bör ge eleverna motivering till varför så stor vikt läggs vid systemering, samt en överblick över systemutvecklingsarbetets plats i samhälle och företag. Sålunda är det t ex ur facklig synpunkt viktigt med en enhetlig terminologi och en formalisering av systemarbetet för möjligheten att påverka utformningen av ett databehandlingssystem. Genomgången bör begränsas till begrepp som behövs för den översiktliga förståelsen av systemarbete och för uppstartning av det första projektet. Introduktionen av speciella begrepp och tekniker uppskjuts i största möjliga utsträckning till det styrda projektarbetet på nivå 2, för att där kunna tas upp i sitt logiska sammanhang. Exempelvis kan en modell med etappindelning presenteras översiktligt i ett sammanhang här, men också presenteras etappvis i anslutning till arbetet med det styrda projektet.

Det är viktigt att delmomentet begränsas till det allra nödvändigaste, så att eleverna snarast möjligt får börja arbeta i projekt. Först när de får arbeta praktiskt med problemen börjar de inse vad systemeringsarbete egentligen innebär.

<u>Innehåll</u>	<u>Kommentarer och exempel</u>
5.1 <u>Introduktion och grundbegrepp</u>	
Databehandlingssystem	Orientering om systembegreppet samt någon övning i "systemtänkande" (att beskriva ett system). Orientering om olika typer av databehandlingssystem (informationssystem, bassystem) samt deras samband och funktion i företag.
Administration	Orientering om begreppen funktion och delfunktion (avgränsade arbetsuppgifter) i företag, samt om beslutsprocesser, för att ge en grund för arbetet med analys av informationsbehov.
Administrativ rationalisering	Orientering om olika områden inom administrativ rationalisering med särskild tonvikt lagd på systemrationalisering. Manuell kontra automatisk databehandling.
Krav på systemutvecklingsarbete	Orsaker till olika krav på systemutvecklingsarbetets uppläggning, såsom hänsyn till berördas kunskaper och synpunkter, möjlighet att kontrollera arbetets fortskridande för att eventuellt förändra inriktningen etc. Etapper.

NIVÅ 2. STYRT PROJEKT

Projektarbetet görs gruppvis och bör omfatta en och samma systemeringsuppgift för alla grupper. För att alla elever skall vara så aktiva som möjligt bör grupperna bestå av högst två-fyra elever. Arbetet inriktas på att ge eleverna dels erfarenhet av hur ett systemutvecklingsprojekt kan läggas upp och genomföras i stort, dels kunskap om arbetsmetoder och tekniker under de olika etapperna.

Stark lärarstyrning med ofta förekommande genomgångar och uppföljningar med hela klassen ger tillfälle att steg för steg introducera och kommentera den systemutvecklingsmetod som tillämpas och de tekniker som övas. Oberoende av vilken metod som väljs för projektarbetet bör eleverna orienteras om några andra metoder, i första hand de som nämns här nedan.

Uppgiften bör begränsas. I en systemutvecklingsmodell med etapper bör t ex uppmärksamheten koncentreras mot de etapper som föregår programmering (i RAS-modellen etapperna förundersökning - systemstudie). Dessutom bör man undvika att ta med alltför många olika transaktionstyper.

<u>Innehåll</u>	<u>Kommentarer och exempel</u>
5.2 <u>Systemeringsprojekt</u>	

Ett systemeringsprojekt	Projektet skall ge kunskap om arbetsmetodik och tekniker i projektinriktat systemutvecklingsarbete samt erfarenhet av någon metod för system-
-------------------------	---

	utveckling. Som exempel på lämplig projektuppgift kan nämnas någon form av register, t ex personregister, (medlems-, elev-) eller lagerregister eller ett enkelt bokningssystem.
Arbetsmetodik	Projektorganisation, innefattande definition av projekt, beslutsgrupp - projektgrupp, personal i projektorganisation samt beslutsgruppens uppgifter. Initiering av projekt. Grovplanering av projekt, inklusive etappindelning. Gemensamma projektadministrativa krav för varje etapp.
Tekniker	Framtagning av dokumentation i anslutning till olika etapper i systemutvecklingsarbetet. Som exempel kan nämnas målbeskrivning och kravspecifikation. Vidare kan förekomma: <ul style="list-style-type: none"> - Framtagning av ut- och indata till systemet. - Analys av samband mellan data och processer i systemet. - Framtagning av dataflödesplaner. - Beskrivning av hur registren skall organiseras. - Framtagning av preliminära programbeskrivningar (in- och utdata till respektive program samt programfunktioner). - Beskrivningar av poster, ingående i registren. - Beskrivningar av layouter för skärmbilder och listor.
Systemutvecklingsmetoder	Här kan tas upp: <ul style="list-style-type: none"> - Enkel output-inputanalys. - Någon eller några modeller med etappindelning, t ex SIS-RAS, ISAC. - Experimentell systemutveckling, t ex systemskisser med APL, prototypsystem med CS4.

NIVA 3. FRITT PROJEKT

Nedanstående kommentarer gäller rena systemeringsprojekt. I de fall man arbetar med fullständiga projekt (systemering och programmering) gäller kommentarerna systemeringsdelen (fram till och med definitionen av programförutsättningar). Kommentarer till programmeringsdelen av ett fullständigt projekt återfinns under moment 2, nivå 3.

Syftet med projektarbetet är att eleverna skall tillämpa och fördjupa kunskaperna i systemutvecklingsarbete samt öva projektarbete i arbetslag.

Av flera skäl är det lämpligt att låta alla grupperna arbeta med en och samma projektuppgift. Det underlättar för såväl läraren som eleverna, om läraren kan välja ut och arbeta igenom en uppgift som är anpassad till elevernas mognad och erfarenheter. Läraren vet i förväg var eleverna upplever speciella svårigheter och kan förbereda sig på hur de kan vägledas mot rimliga lösningar och vid gemensamma redovisningar känner eleverna igen problemen och har lätt att följa med i de andra gruppernas resonemang.

För lärare med stor erfarenhet av undervisning i systemering kan det vara stimulerande att låta elevgrupperna arbeta med sådana projekt. För eleverna innebär det i så fall en extra stimulans om projektuppgifterna hämtas från omgivningen och kan leda till mål som nås mer lättare än vid ett vanligt undervisningsmoment. Den direkta kontakten med användarna för framtagna krav på systemet kan ge eleverna många värdefulla erfarenheter. Denna typ av undervisningsmodell ställer dock stora krav på lärarens kunskaper, erfarenhet och engagemang. Det finns risk för att elevernas behållning, i fråga om kunskaper och färdigheter i genomförande av systemutvecklingsprojekt, inte blir likvärdig med den, som de får om modellen med en enda projektuppgift används. Det kan också vara risk för att projekten urartar till rena programmeringsprojekt.

Momentet systemering upplevs som svårt. Valet av projektuppgift har stor betydelse här. Sådana kan det vara svårt att arbeta med projekt inom områden, som innehåller alltför många, för eleverna oberoanta begrepp.

Det är vanligt att systemeringsprojekt blir mer arbetskrävande än man från början inser. Därför är det viktigt att, trots den relativt väl tilltagna tiden, läraren ser till att uppgiften begränsas. Detta bör ske genom en hård avgränsning av systemet till ett litet antal delsystem och genom begränsning av antal transaktionstyper, samt genom att projektarbetet koncentreras på systemutvecklingsstegen före programmering. Läraren bör uppmärksamma detta redan vid val och fördelning av uppgifterna, men även fortlopande under arbetets gång, t ex genom projektplans- och etappredovisningar (se nedan).

Projektet bör planeras och genomföras enligt en uppgjord plan med etappredovisningar och beslutspunkter. Krav på arbetsmetodik och dokumentation enligt förut svade principer bör ställas.

Innehåll

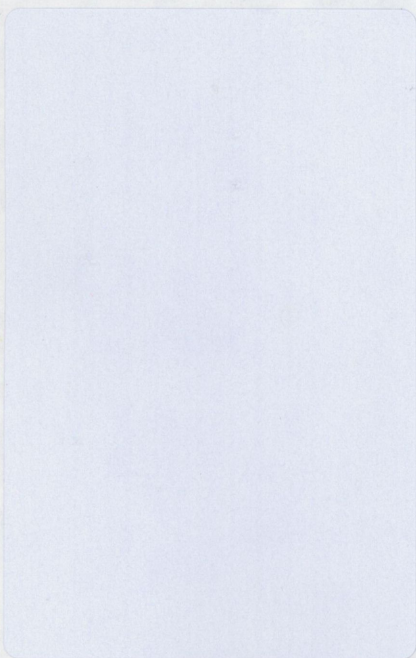
Kommentarer och exempel

5.2 Systemeringsprojekt

Ett systemeringsprojekt	<p>Arbetet utförs i grupp. Gruppstorleken anpassas till elevernas förutsättningar och den målsättning man har med gruppens arbete. I små arbetslag (2-4 elever) tvingas varje elev att mer aktivt ta del i de flesta av projektets olika delaktiviteter. I stora arbetslag (5-8 elever) får eleverna mer erfarenhet av projektledning och av de speciella planerings- och samordningsproblem som kan uppkomma i projektarbete med många deltagare. Varje grupp genomför ett enda projekt, lika för alla grupper eller olika för skilda grupper. Arbetet bör omfatta minst en tredjedel av den totala tiden för kursen i systemering.</p>
	<p>Exempel på projektuppgifter:</p> <ul style="list-style-type: none"> - Två uppgifter som utprovats och visat sig lämpliga är prenumerationsregister för en tidning och medlemsregister för en förening. - Uppgifter med anknytning till omgivningen kan t ex hämtas från skolans expedition (uppföljning av skolans budget) eller bibliotek (uppföljning av boklån), från kommunens skolmåtidsavdelning (matsedel, skolmåtidsplanering), från andra kommunala förvaltningar (t ex planering av en statistisk undersökning) eller från enskilda företag.

GÖTEBORGS
UNIVERSITETSBIOTEK
BIBLIOTEKET I MÖLNDAL

**TILLHÖR REFERENSBIBLIOTEKET
UTLÄNAS EJ**



Läroplan för gymnasieskolan

Lgy⁷⁰

II Supplement 95