

# Yttrande om Skolverkets förslag ”Alla gymnasieelever ska kunna läsa programmering”

Diarienummer: 2016:01482

---

*Sammanställt av: Stefan Andersson (Saab AB), Diana Unander Nordle (LNU), Fredrick Rybarczyk (Patwick), Örjan Johansson (Atea), Fredrik Heintz (LiU)*

*Kommenterat och underskrivet av: se sid 3.*

## Sammanfattning

Skolverket har från regeringen fått ett uppdrag att stärka den digitala kompetensen i skolan. Målet är bl. a att alla gymnasieelever som går ett nationellt program ska kunna läsa programmering. Skolverket har därför tagit fram förslag till ändringar i Skolverkets föreskrifter för att möta detta mål. Totalt sett tycker vi att det är en bra inriktning och ett steg i rätt riktning: att få in programmering i gymnasieskolan och på Komvux är av stor vikt för att Sverige ska vara konkurrenskraftigt även i framtiden. Digitaliseringen av Sverige påverkar alla branscher och yrkesgrupper, och ställer nya krav på medborgarna. Samtidigt har vi en ojämn fördelning mellan tjejer och killar som läser vidare och fördjupar sig inom digital teknik.

Vi vill se en tydligare kravställning att de grundläggande kurserna ej kan väljas bort – varken av eleven eller av skolan. Vi kan inte säga att det är viktigt och samtidigt ge för stor frihet att välja bort ämnena. En annan synpunkt är vikten av att få in roliga uppgifter och små tillämpningsprojekt i kurserna: det är viktigt att det inte upplevs som att programmering är ett självändamål. Progressionen från grundskolan bör framgå tydligare. Initialt så kommer kurserna behöva vara på grundläggande nivå då eleverna kommer helt utan kunskaper från grundskolan. I ett längre perspektiv så bör det dock finnas en tydlig progression då programmering och digital teknik kommer att bli tydligare även i tidiga år. Vi har fått in feedback från lärare på gymnasiet som vittnar om att elevernas förmåga till att hantera en dator som ett arbetsverktyg snarare än en app-uppbyggd plattform (mobil, platta) har försämrats betydligt under senare år. Många elever saknar helt kunskap och förståelse för enkla saker såsom att spara ner dokument i ett mappsystem, detta måste tas i beaktande.

Kommentarer som inte direkt berör remissen men som har en nära koppling. Hur skall lärarnas kompetens säkerställas. Kurserna beräknas träda i kraft inom en snar framtid på bred front samtidigt som lärare med rätt kompetens saknas. Detta gäller dels den grundläggande kompetensen att ge kurserna men då tekniken är under ständig och snabb utveckling så måste det också finnas tydliga planer/strategier för att säkerställa lärarnas kontinuerliga kompetensutveckling. Remissförslaget saknar också helt koppling till hur

näringslivets kompetenser kan nyttjas här, både direkt i mötet med elever men också koppla till lärarnas kompetensutveckling.

## Innehåll

1	Allmänna synpunkter.....	4
1.1	Viktigt för genomförandet av förändringen.....	4
1.2	Valbarhet kontra kravställning.....	4
1.3	Nödvändiga färdigheter för yrkeskarriär efter gymnasiet.....	5
1.4	Nödvändiga färdigheter för fortsatta studier efter gymnasiet.....	6
1.5	Påverkan på lärarutbildningen.....	7
2	Ämnesplaner.....	8
2.1	Datalagring.....	8
2.2	Gränssnittsdesign.....	8
2.3	Mjukvarudesign.....	9
2.4	Mobila applikationer.....	9
2.5	Programmering.....	10
2.6	Tillämpad programmering.....	11
2.7	Webbserverprogrammering.....	11
2.8	Webbutveckling.....	12

## Kommenterat och underskrivet av

**Monica Almqvist**, Docent och föreståndare för Vattenhallen Science Center, LTH  
**Rikard Andersson**, SW Development Researcher, Ericsson AB  
**Stefan Andersson**, Programledare Framtida Flygsystem, Saab AB  
**Jan Carlson**, Professor i datavetenskap, Mälardalens Högskola  
**Anders Caspár**, Director External Research Relations, Ericsson AB  
**Johan Ellborg**, Lösningsspecialist Utbildning, Microsoft  
**Tor Ericson**, Manager Electronics Design, ÅF  
**Fredrik von Essen**, Näringspolitisk expert, IT&Telekomföretagen  
**Veronica Gaspes**, Lektor i datavetenskap, Högskolan i Halmstad  
**Anders Gerestränd**, Universitetsadjunkt, Linnéuniversitetet  
**Fredrik Heintz**, Docent i datalogi, Linköpings universitet  
**Helena Holmström Olsson**, Associate professor, Malmö Högskola  
**Örjan Johansson**, Nationell Affärsenhetschef, Atea  
**Johan Leitet**, Universitetsadjunkt, Linnéuniversitetet  
**Niklas Lindhardt**, Partner, CaptureInnovation AB  
**Kurt-Lennart Lundbäck**, VD och grundare, Arcticus Systems AB  
**Kristina Lundqvist**, Professor Dependable Software Engineering, Mälardalens Högskola  
**Birgitta Lundvik**, VD, Hansoft Technologies AB & Favro AB  
**Henrik Lönn**, Technology Specialist, Embedded Software Development, Volvo Group  
**Claes Magnusson**, Rektor, Malmö Yrkeshögskola  
**Michael Mattsson**, Professor i programvaruteknik, Blekinge Tekniska Högskola  
**Marcelo Mildrad**, Professor, Linnéuniversitetet  
**Gabriel Modéus**, Generalsekreterare, Swedsoft  
**Annika Mårtensson**, Prorektor, Lunds Tekniska Högskola  
**Diana Unander Nordle**, Forsknings- och projektkoordinator, Linnéuniversitetet  
**Ingrid Nordmark**, VD, RISE SICS AB  
**Björn Regnell**, Professor Software Engineering, Lunds Tekniska Högskola  
**Peter Roos**, VD, FindOut Technologies AB  
**Mikael Sjödin**, Professor och Forskningsledare, Mälardalens Högskola  
**Martin Törngren**, Professor, KTH  
**Jonas Wallberg**, Director ICT, Teknikföretagen  
**Claes Wohlin**, Professor och dekan, Blekinge tekniska högskola  
**Anders Åström**, Affärsområdeschef Industri, Combitech AB

## 1 Allmänna synpunkter

### 1.1 Viktigt för genomförandet av förändringen

#### 1.1.1 Kommentarer

- Föreslagna förändringar kommer att ske stegvis. Det innebär, anser vi, att Skolverket behöver en tydlig (och explicit) plan för hur detta skall hanteras/planeras: t ex hur man skall hantera skillnaden mellan de som i början bara läst programmering i 9:e klass och kommer till gymnasiet 2020 jämfört med de som tio år senare kommer till gymnasiet och har läst programmering ända sedan förskolan. Detta borde rimligen påverka hur undervisningen utformas så att den anpassas till en stor variation av förkunskaper över tid.
- Skolverket bör samordna sina insatser med utbildningsdepartementet så att en kraftfull och långsiktig utveckling av lärarhögskolornas verksamhet och forskning inom datavetenskaplig didaktik planeras och finansieras. Ämnet datavetenskap bör finnas med i kursplaner på lärarutbildningen och doktorandtjänster för forskning inom datavetenskaplig didaktik med fokus på didaktik i grundskola och gymnasium behöver inrättas.

#### 1.1.2 Förslag till ändringar

- Skolverket behöver ta fram en utvecklingsplan för digitaliseringen av skolan (införandet av programmering) som omfattar fler år än bara 2017-2018.
- Motsvarande plan för utvecklingen av lärarhögskolan (inkluderande bl. a de förslag på förändringar som lyfts fram i kapitel 1.5.2 i detta remissvar) behöver tas fram av skolverket och utbildningsdepartementet tillsammans.

### 1.2 Valbarhet kontra kravställning

#### 1.2.1 Kommentarer

- Skolverkets förslag har som mål att alla gymnasieelever ska ha möjlighet att lära sig programmering – det är bra. Det är dock oklart vilken nivå som är krav för respektive program. Kurserna Programmering 1 och Tillämpad programmering ligger under programfördjupning. Det innebär att det är frivilligt för eleven att välja dessa kurser. Det är även frivilligt för skolorna att erbjuda kurserna.
- Vi anser att det behövs en uttalad lägsta nivå för följande program för att bädda för vidare studier och möta kunskapsbehov kopplat till den ökade digitaliseringen inom ekonomi, natur- och samhällsvetenskapen samt inom ingenjörsvetenskapen:
  - Ekonomiprogrammet
  - Naturvetenskapsprogrammet
  - Samhällsvetenskapsprogrammet
  - Teknikprogrammet

- Att säkra en grundnivå kunskapsmässigt inom programmering skulle också bidra till att minska dagens problem i ingenjörsutbildningarna med studenter som har mycket olika förkunskaper. Rekryteringen till civilingenjörsutbildningarna sker t ex främst genom naturvetenskapsprogrammet och det är därför extra viktigt för de tekniska högskolorna att dessa studenter har en solid grund i programmering. De gymnasieskolor som erbjuder naturvetenskapliga program behöver därför, anser vi, ha obligatorisk programmering med flera påbyggnadskurser så att djupet blir större.
- Vi anser dessutom att det behövs en uttalad lägsta nivå även för följande yrkesutbildningar:
  - Industritekniska programmet – för att möta digitaliseringen av industrin
  - El- och energiprogrammet – för att möta digitaliseringen inom el- och energisektorn
  - Barn- och fritidsprogrammet – för att kunna arbeta med programmering med barn och unga i sina kommande yrkesliv
- Vidare bör Skolverket överväga hur gymnasiet kan bidra till att fler kvinnliga studenter väljer att studera på Datavetenskaps- och Elektroteknik-program på landets högskolor. Enligt ny forskning är det mycket viktigt med positiva förebilder, social tillhörighet och gott självförtroende om ämnesområdet när kvinnor (och män) gör sina utbildningsval, ref Una Tellhed, Martin Bäckström & Fredrik Björklund (2016).

### 1.2.2 Förslag till ändringar

- Gör följande kurser obligatoriska för Ekonomi-, Naturvetenskaps-, Samhällsvetenskaps- och Teknikprogrammen, samt yrkesutbildningarna Industritekniska programmet, El- och energiprogrammet samt Barn- och fritidsprogrammet:
  - Programmering 1
  - Tillämpad programmering
- De obligatoriska kurserna skall dock ta hänsyn till vad eleverna kan från grundskolan. De obligatoriska kurserna får inte upplevas som ett steg bakåt för eleverna. Det borde kunna hanteras genom att kurserna får roliga och utmanande problem att lösa med programmering.

## 1.3 Nödvändiga färdigheter för yrkeskarriär efter gymnasiet

### 1.3.1 Kommentarer

- Utöver frågan om att vissa kurser bör vara obligatoriska precis som svenska och matematik, så anser vi att man behöver ha en tydlig målbild för vilka grundläggande färdigheter som de olika kurserna i de olika programmen skall leda till. Detta för att bädda för en ökad flexibilitet och mobilitet på

arbetsmarknaden.

Kravbilden måste vara tydlig i examinationskraven så att man säkrar att man uppnår mål enligt nedan.

- Vi anser att de som lämnar gymnasiet för att direkt komma ut i arbetslivet behöver följande grundläggande färdigheter avseende programmering och digital kompetens:
  - Grundläggande förståelse för datorer, operativsystem och nätverk
  - Grundläggande förståelse för programmering och datalogiskt tänkande, dvs kunna skriva kortare egna program utifrån en problembeskrivning samt testa och dokumentera det för att göra det förståeligt för utomstående. Förmåga att använda vanliga konstruktioner på ett lämpligt sätt och bryta ner problemet i lagom stora och väldefinierade delar. Namnge och parameterisera funktioner på ett för uppgiften lämpligt sätt samt använda biblioteksfunktioner på ett adekvat sätt.
  - God insikt om etiska och säkerhetsmässiga aspekter på den digitala tekniken och hur man skall förhålla sig till dessa (uppförande, kritiskt tänkande och säkerhetstänkande)
  - Grundläggande förståelse för systematiskt arbetssätt för att lösa problem med digital teknik (problemformulering, analys, design, implementering och test)

### **1.3.2 Förslag till ändringar**

- Säkerställ att förslag till obligatoriska kurser i gymnasiet enligt kapitel 1.1.2 ger dessa färdigheter.

## **1.4 Nödvändiga färdigheter för fortsatta studier efter gymnasiet**

### **1.4.1 Kommentarer**

- Utöver frågan om att vissa kurser bör vara obligatoriska precis som svenska och matematik, så anser vi att man behöver ha en tydlig målbild för vilka grundläggande färdigheter som de olika kurserna i de olika programmen skall leda till. Detta för att bädda för att underlätta fortsatta studier på universitetet.
- Vi anser att de som lämnar gymnasiet för att fortsätta med studier på universitetet behöver följande grundläggande färdigheter avseende programmering och digital kompetens:
  - Grundläggande förståelse för datorer, operativsystem och nätverk.
  - Grundläggande förståelse för programmering och datalogiskt tänkande, dvs kunna skriva kortare egna program utifrån en problembeskrivning samt testa och dokumentera det för att göra det förståeligt för utomstående. Förmåga att använda vanliga konstruktioner på ett lämpligt

sätt och bryta ner problemet i lagom stora och väldefinierade delar. Namnge och parameterisera funktioner på ett för uppgiften lämpligt sätt samt använda biblioteksfunktioner på ett adekvat sätt.

- God insikt om etiska och säkerhetsmässiga aspekter på den digitala tekniken och hur man skall förhålla sig till dessa (uppförande, kritiskt tänkande och säkerhetstänkande).
- Grundläggande förståelse för systematiskt arbetssätt för att lösa problem med digital teknik (problemformulering, analys, design, implementering och test).
- Grundläggande förmåga att använda modellerings- och analysverktyg t ex för att modellera fysikaliska förlopp och för att analysera data.
- Har provat att arbeta iterativt och experimenterade, s.k. agilt.

#### **1.4.2 Förslag till ändringar**

- Säkerställ att förslag till obligatoriska kurser i gymnasiet enligt kapitel 1.1.2 ger dessa färdigheter.

### **1.5 Påverkan på lärarutbildningen**

#### **1.5.1 Kommentarer**

- Viktigt att digitaliseringen genomsyrar hela lärarutbildningen eftersom denna transformation påverkar alla ämnen och alla framtida yrken. Har vi en lärarkår som är införstådd med det tror vi att chansen att nå ut till eleverna ökar markant.
- Viktigt att komplettera lärarnas grundutbildning möjligheter till fortlöpande fortbildning under hela lärarkarriären.
- Möjligheterna att använda yrkeshögskolor som alternativ till kompetenslyft av lärare bör också övervägas.
- Vår bedömning är att det behövs större flexibilitet och bättre möjligheter för universitetsstudenter att mixa kurser från traditionell lärarutbildning och kurser inom datavetenskaps-/dataingenjörsprogram.
- Viktigt att så långt som möjligt öppna upp för studenter att efter examen kunna välja mellan läraryrket och ingenjör, och sedan kunna växla mellan dessa karriärer.

#### **1.5.2 Förslag till ändringar**

- Kräv att alla nyutbildade lärare har programmeringskunskaper åtminstone motsvarande programmering 1 då programmering numera ingår i grundskolan.

- Gör det möjligt för ingenjörstudenter att välja kurser från lärarprogrammet, och gör kurser inom ingenjörsprogram (t ex datorteknik, matematik mm) tillgängliga för lärarstudenter.
- Gör det möjligt, och till och med vanligt, att studenter samtidigt kan ta dubbla examen: lärarbehörighet och ingenjörsexamen.
- Fler kurser än de som remissas här bör genomlysas med syfte att få in digitaliseringens konsekvenser (möjligheter).
- Kanske man regionalt kan använda initiativ som t ex Kodcentrum (<https://www.kodcentrum.se/>) eller Coder Dojo (<http://coderdojo.se/>) anpassade för lärare, som därmed kan ges möjlighet att tränas i programmering med samma infrastruktur. Kanske i samverkan med lokala mjukvaruföretag och akademi.

## 2 Ämnesplaner

### 2.1 Datalagring

#### 2.1.1 *Kommentarer*

- Väldigt specifik kurs som riskerar att bli för avgränsad och svår att använda utanför skolan utan en bra kontext och bra tillämpningsexempel
- Det är viktigt att konkretisera de teoretiska delarna. Kursen bör vara ett bra underlag för att kunna arbeta vidare med datalagring såväl praktiskt som teoretiskt.
- Begrepp som ontologier och informationsmodellering saknas

#### 2.1.2 *Förslag till ändringar*

- Säkerställ att kursen omfattar datalagring i en kontext: vad, hur och varför.
- För in begrepp som ontologier och informationsmodellering
- Inför projektarbete/uppgift att lösa praktiska problem som kräver lösningar för datalagring och åtkomst

### 2.2 Gränssnittsdesign

#### 2.2.1 *Kommentarer*

- Kursen har ett för stort fokus på själva gränssnittet mellan webb-applikationer och människa i traditionell bemärkelse.

#### 2.2.2 *Förslag till ändringar*

- Problematik och lösningar för interaktion mellan inbyggda smarta (AI) system inkl. självlärande system och människa bör beröras i kursen.



## 2.3 Mjukvarudesign

### 2.3.1 Kommentarer

- Innehållet verkar relevant för denna typ av kurs, men omfattningen kan ses som stor. Det kan finnas behov av fokusering på något redskap för mjukvarudesign.
- Vad avses med ”Förmåga att skapa internationaliserade applikationer och att lokalisera applikationer”?
- Ämnets syfte Punkt 6 ”Kunskaper om arbetsmiljöfrågor” är relevant vad gäller arbetslivet generellt, men har inte med mjukvarudesign att göra. Förvirrar mer än bidrar.
- Begrepp som ”arkitektur”, ”modularisering” och ”partitionering” verkar ej behandlas

### 2.3.2 Förslag till ändringar

- Ersätt ”internationaliserade applikationer” med ”FOSS, Fri programvara och programvara med öppen källkod”. FOSS är idag ett väletablerat arbetssätt där man använder andras program som delar i sina egna applikationer, och att man regelmässigt bidrar tillbaka till internationella forum för olika typer av FOSS.
- Ersätt Punkt 6 under ”Ämnets syfte” med en punkt om ”arkitektur”

## 2.4 Mobila applikationer

### 2.4.1 Kommentarer

- Denna kurs hanterar tekniker som är väldigt föränderlig (utveckling för mobil/webb).
- Med tanke på de olika kraven som olika mobiltillverkare ställer kan det finnas problem i att fullt ut täcka de utvecklingsmiljöer som står till buds.
- För specifik kurs – borde istället vara en kurs för utveckling av programvara för många olika plattformar och operativsystem: mobiler, surfplattor, iOS, Android, Linux, Unix, Windows etc.

### 2.4.2 Förslag till ändringar

- Viktigt att kursen inte blir för specifik när det gäller teknik. Även viktigt med fortlöpande kompetensutveckling av lärare bör tas om hand. Detta så att kurserna inte snabbt blir föråldrade.
- Formulera om kursens syfte till att den ska ge färdigheter i att utveckla mjukvaruapplikationer för olika typer av plattformar.

## 2.5 Programmering

### 2.5.1 Kommentarer

- Uppdelningen i de olika delkurserna (P1-P3) ger en bra möjlighet till att separera omfattningen och innehållet på ett rimligt sätt.
- Kurserna har ett för stort fokus på aktiviteten att skriva program i förhållande till hur man tänker för att skapa program. Detta finns dock omnämnt i kursen Tillämpad programmering.
- Kurserna verkar vara för omfattande. Det är naturligtvis omöjligt att bedöma exakt bredd och djup från de beskrivningar som finns tillgängliga.
- Viktigt att inte kurserna framstår som ”programmering för programmeringens egen skull”. Viktigt att elevernas intresse väcks genom att de får uppdrag att utföra, lösningar som ska tas fram, saker som ska hända eller ändras.
- Ämnets syfte Punkt 1: Vad innebär ”Kunskaper om programmerbara system”?
- Ämnets syfte Punkt 2: Bör omformuleras – ger felaktigt fokus och saknar koppling till modellbaserad utveckling
- Ämnets syfte Punkt 9: Bör ses över – ”validering” saknas
- Agil utveckling, dvs iterativ och experimenterande programvaruutveckling som uppmuntrar att man involverar de tänkta användarna.
- Områden som programmering av smarta (autonoma) och/eller självlärande system saknas i utbudet. Dessa områden håller idag på att bli en realitet i stort sett alla branscher och sektorer.

### 2.5.2 Förslag till ändringar

- Komplettera med moment som hanterar hur man tänker och arbetar när det gäller att skapa program.
- Förtydliga att fokus är eleverna skall få möjlighet att lösa problem med hjälp av programmering och det skall leda till synliga och roliga resultat, t ex programmering av bilar eller robotar.
- Se över kurserna så att omfattningen blir rimliga för lärare och elever.
- Formulera om Ämnets syfte Punkt 1 till: ”Kunskaper om olika typer av programvarusystem: öppna, inbyggda, säkerhetskritiska, självlärande etc”.
- Formulera om Ämnets syfte Punkt 2 till: ”Modellering med UML/SysML och andra notationer”.
- Formulera om Ämnets syfte Punkt 9 till: ”Testmetodik, verifiering och validering av programvarusystem”.

- Lägg till en Punkt 10 under Ämnets syfte: ”Agil (iterativ och experimenterande) programvaruutveckling”.
- Överväg att lägga in AI-programmering t ex i form av självlärande/autonoma system i den mer avancerade programmeringskursen.

## 2.6 Tillämpad programmering

### 2.6.1 Kommentarer

- Ämnets syfte Punkt 2 ”Kunskaper om utvecklingsprocessen inom programmeringsområden och förståelse av sambanden mellan de olika delarna i den.” saknar tydlig koppling till systemkontext (eller ett s.k. systemperspektiv).
- Ämnets syfte Punkt 4 ”Förmåga att analysera konsekvenserna av programmeringens betydelse för individ, samhälle och teknisk utveckling utifrån hållbarhet, etik och genus.” är väldigt krystad. Etik och genus-aspekter bör behandlas i andra kurser.
- Kursen har en bred omfattning av tillämpningsområden i sin nuvarande definition. Trots det nämns vare sig mekatronik eller inbyggda system.
- På vissa program har de läst programmering 1, medan på andra program har de läst flera olika kurser med programmering innan de börjar. Det innebär att förkunskaperna kommer vara olika på de elever som börjar kursen.

### 2.6.2 Förslag till ändringar

- Fundera på att komplettera med material om arbetsmetodik och hur arbete i grupp organiseras, om detta inte föregått kursen i andra delar.
- Formulera om Ämnets syfte Punkt 2 till ”Kunskaper om processer för utveckling av programvara med ett systemperspektiv”.
- Ersätt Punkt 4 under Ämnets syfte med ”Kunskaper och färdigheter i att hantera egenskaper som säkerhet, testbarhet, utvecklingsbarhet och prestanda”.
- Lägg till programmering av inbyggda system för styrning av robotar och andra mekatroniska system i beskrivningen av innehållet i kursen. Uppmuntra laborering och experimentering.

## 2.7 Webbserverprogrammering

### 2.7.1 Kommentarer

- Viktigt att trycka på cyber-säkerhet och hur man bygger robusta system

### 2.7.2 Förslag till ändringar

- Förtydliga så att ovan tas om hand.

## **2.8 Webbutveckling**

### **2.8.1 Kommentarer**

- Viktigt att trycka på användargränssnitt respektive robusthet, dvs hur man bygger användbara / lättbegripliga och robusta Webb-applikationer.
- Ambitionsnivån är hög

### **2.8.2 Förslag till ändringar**

- Förtydliga så att ovan tas om hand.
- Undvik ”val av webbhotell”. Ändra till ”val av driftsmiljö