

Programmering

Programmering har stor betydelse för och påverkan på arbetslivet och samhället. Ämnet programmering behandlar hur mjukvaror skapas, anpassas och utvecklas samt programmeringens roll i informationstekniska sammanhang som datorsimulering och praktisk datoriserad problemlösning. Ämnet behandlar även hur programmerade system påverkar maskiner, utrustning och beslutsfattande.

Ämnets syfte

Undervisningen i ämnet programmering ska syfta till att eleverna utvecklar kunskaper om programmeringens grunder, färdigheter i att tillämpa relevanta metoder för programutveckling och förmåga att lösa programmeringstekniska problem. Undervisningen ska också leda till att eleverna utvecklar färdigheter i programvaruutveckling och förståelse av hur programvaror och programmerbara system påverkar sin omgivning.

Undervisningen ska bidra till att eleverna utvecklar förmåga att analysera, designa och vidareutveckla program. Undervisningen ska också bidra till att eleverna utvecklar kunskaper om några vanligt förekommande programspråk samt traditioner och nya trender i utvecklingen av programspråk. Eleverna ska även ges möjlighet att utveckla förståelse för hur programmering används utifrån olika sociala aspekter, som exempelvis genus, kultur och socioekonomisk bakgrund. Dessutom ska undervisningen leda till att eleverna utvecklar kunskaper om datorns användning i samhället och datorns möjligheter och begränsningar.

I undervisningen ska eleverna ges möjlighet att arbeta i projekt och att enskilt eller i grupp utföra programmeringsuppgifter av varierande komplexitet och inom olika tillämpningsområden. Undervisningen ska lämna utrymme för diskussion om och reflektion över etiska frågor och olika handlingsalternativ inom programmering.

Undervisningen i ämnet programmering ska ge eleverna förutsättningar att utveckla följande:

- Kunskaper om programmerbara system, programspråk och programmeringsparadigm.
- Förmåga att formulera och skapa lösningar för programmeringsproblem.
- Förmåga att skapa program med anpassning till syfte, användare och tillgänglighet.

Nivåer i ämnet programmering

- Nivå 1, 100 poäng.
- Nivå 2, 100 poäng, som bygger på nivå 1.

Nivå 1, 100 poäng

Nivåkod: PROG1000X

Centralt innehåll

Undervisningen i ämnet programmering på nivå 1 ska behandla följande centrala innehåll:

System

- Programmeringens möjligheter och begränsningar utifrån datorns funktionssätt.
- Samspelet mellan program, exekveringsmiljö, operativsystem och hårdvara.
- Användning av datavetenskapliga begrepp och principer.

Problemlösning

- Strukturerad problemlösning och programmering.
- Tillämpning av någon metod för att underlätta problemlösning, till exempel pseudokod och diagramteknik.

Programmering

- Programmering i ett eller flera programspråk, däribland minst ett textbaserat.
- Användning av relevanta verktyg för programutveckling, till exempel IDE (integrated development environment) och versionshanteringssystem.
- Det valda programspråkets grundläggande kontrollstrukturer.
- Skapande av enkla program som löser någon uppgift.
- Olika sociala aspekter av hur programmering används.
- Enklare metoder för att finna, analysera, åtgärda, förebygga och förhindra syntaxfel, programmeringslogiska fel och exekveringsfel.
- Enklare undantagshantering.
- Goda principer för namngivning, kommentarer och strukturerad kod.
- Gränssnitt för interaktion mellan program och användare.

Nivå 2, 100 poäng

Nivåkod: PROG2000X

Centralt innehåll

Undervisningen i ämnet programmering på nivå 2 ska behandla följande centrala innehåll:

System

- Det objektorienterade programmeringsparadigmet.

Problemlösning

- Analys, nedbrytning och modellering av programmeringstekniska problem med lämpligt analysverktyg, till exempel användningsfall. Design av lämplig lösning utifrån gjord analys med lämpliga verktyg och metoder.

Programmering

- Programmering i objektorienterat programspråk.
- Användning av relevanta verktyg för programutveckling, till exempel modelleringsverktyg.
- Skapande och användning av klasser och objekt utifrån gjord analys och design.
- Hur arv används för att förändra eller utöka beteende hos klasser som ingår i egna och andras klasshierarkier och standardbibliotek.
- Användning av generiska klasser och metoder.
- Skrivning och läsning av lagrad data i filer och databaser.
- Utveckling av program som nyttjar kommunikation över internet.
- Metoder för att förebygga, finna, analysera, åtgärda och förhindra syntaxfel, programmeringslogiska fel och exekveringsfel.
- Testning och undantagshantering.
- Normer och värden inom programmering, till exempel läsbarhet, dokumentation, testbarhet och nyttan av kodstandard.
- Strukturering av programkod och klasshierarkier.
- Skapande av användarvänliga gränssnitt.

Betygskriterier

Av 15 kap. 24 § andra stycket och 20 kap. 37 § andra stycket skollagen (2010:800) följer att läraren vid betygssättningen i ett ämne ska göra en sammantagen bedömning av elevens kunskaper på den aktuella nivån i ämnet i förhållande till de betygskriterier som gäller för ämnet som helhet och sätta det betyg som bäst motsvarar elevens kunskaper. Samtliga kriterier för betyget E ska dock vara uppfyllda för att eleven ska kunna få ett godkänt betyg.

Betyget E

Eleven visar **godtagbara** kunskaper om programmerbara system, programspråk och programmeringsparadigm.

Eleven formulerar och löser **enkla** programmeringsproblem.

Eleven skapar program med **viss** anpassning till syfte, användare och tillgänglighet.

Betyget D

Elevens kunskaper bedöms sammantaget vara mellan C och E.

Betyget C

Eleven visar **goda** kunskaper om programmerbara system, programspråk och programmeringsparadigm.

Eleven formulerar och löser **relativt komplexa** programmeringsproblem.

Eleven skapar program med **god** anpassning till syfte, användare och tillgänglighet.

Betyget B

Elevens kunskaper bedöms sammantaget vara mellan A och C.

Betyget A

Eleven visar **mycket goda** kunskaper om programmerbara system, programspråk och programmeringsparadigm.

Eleven formulerar och löser **komplexa** programmeringsproblem.

Eleven skapar program med **mycket god** anpassning till syfte, användare och tillgänglighet.