

Programmering, datastrukturer och algoritmer

Ämnet programmering, datastrukturer och algoritmer behandlar hur mjukvaror skapas, anpassas och utvecklas samt programmeringens roll i informationstekniska sammanhang som datorsimulering och praktisk datoriserad problemlösning. Ämnet behandlar även hur algoritmer påverkar maskiner, utrustning och beslutsfattande. Ämnet får bara anordnas i vidareutbildning i form av ett fjärde tekniskt år i gymnasieskolan.

Ämnets syfte

Undervisningen i ämnet programmering, datastrukturer och algoritmer ska syfta till att eleverna utvecklar specialiserade kunskaper om och färdigheter i programmering. Eleverna ska ges möjlighet att utveckla förmåga att anpassa programmering efter avsedd användare av det färdiga programmet. Genom undervisningen ska eleverna ges möjlighet att utveckla förmåga att tillämpa, skapa och välja designmönster, algoritmer och datastrukturer med hänsyn till såväl önskad funktionalitet som prestanda och säkerhet. Undervisningen ska även ge eleverna möjlighet att utveckla förmåga att skapa feltoleranta program och att lösa programmeringstekniska problem.

Undervisningen ska bidra till att eleverna utvecklar kunskaper om vanligt förekommande programspråk och programmeringsparadigm. Eleverna ska även ges möjlighet att utveckla förmåga att använda datakällor för datautbyte.

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

Undervisningen i ämnet programmering, datastrukturer och algoritmer ska ge eleverna förutsättningar att utveckla följande:

- Kunskaper om programmeringsparadigm och programspråk.
- Förmåga att välja och använda designmönster, algoritmer och datastrukturer.
- Förmåga att skapa program med ett givet syfte och för en avsedd användare eller uppdragsgivare.
- Förmåga att nyttja datautbyte med datakällor samt identifiera, åtgärda och förhindra fel vid programmering och exekvering.

Nivåer i ämnet

Nivå 1, 100 poäng.

Nivå 2, 100 poäng.

Vidareutbildning i form av ett fjärde tekniskt år i gymnasieskolan

Nivå 1, 100 poäng

Nivåkod:

Centralt innehåll

Undervisningen i ämnet programmering, datastrukturer och algoritmer på nivå 1 ska behandla följande centrala innehåll:

- Programmering i objektorienterat språk samt dess egenskaper och användningsområden i förhållande till andraspråk.
- Vanligt förekommande designmönster av enklare karaktär för skapande, struktur och beteende.
- Datastrukturer, däribland för hantering av mängder.
- Vanligt förekommande algoritmer för aktuellt problemområde.
- Värdering av algoritmer och datastrukturers lämplighet i olika situationer.
- Skapande av program som nyttjar klasser, objekt, interface, templates, inkapsling och polymorfism.
- Omstrukturering av kod (refactoring) för att utveckla eller anpassa befintliga program.
- Synlighet och livslängd hos identifierare och variabler.
- Stark och svag samt statisk och dynamisk typning.
- Datautbyte med datakällor och tjänster.
- Permanent lagring av data.
- Metoder för att förebygga och förhindra uppkomst av fel under exekvering.
- Hantering av uppkomna fel under exekvering.
- Avläsning av program för att hitta fel och brister med hjälp av verktyg i aktuell utvecklingsmiljö.

Nivå 2, 100 poäng

Nivåkod:

Centralt innehåll

Undervisningen i ämnet programmering, datastrukturer och algoritmer på nivå 2 ska behandla följande centrala innehåll:

- Olika programmeringsparadigms historik, egenskaper och användningsområden.
- Val av språk efter problemområde.
- Designmönster och mjukvaruarkitektur.
- Analys och val av datastrukturer och algoritmer utifrån problemområde och prestanda.
- Programmering i språk från flera paradigmen.
- Användning av databashanterare för lagring och manipulering av data.
- Användning av webbtjänster, protokoll, gränssnitt och format för datautbyte.

Vidareutbildning i form av ett fjärde tekniskt år i gymnasieskolan

- Kvalitetssäkring och testning av källkod genom manuella och automatiserade metoder, till exempel kodgranskning och enhetstestning.

Betygskriterier

Betyget E

Eleven för **övergripande** resonemang om egenskaper hos samt för- och nackdelar med olika programmeringsparadigm och programspråk.

Eleven väljer och använder med **säkerhet** designmönster, algoritmer och datastrukturer.

Eleven skapar **fungerande** program med ett givet syfte och för en avsedd användare eller uppdragsgivare.

Eleven nyttjar med **säkerhet** datautbyte med datakällor och identifierar, åtgärdar och förhindrar fel vid programmering och exekvering på ett **fungerande** sätt.

Betyget D

Elevens kunskaper bedöms sammantaget vara mellan C och E.

Betyget C

Eleven för **utvecklade** resonemang om egenskaper hos samt för- och nackdelar med olika programmeringsparadigm och programspråk.

Eleven väljer och använder med **god säkerhet** designmönster, algoritmer och datastrukturer.

Eleven skapar **ändamålsenliga** program med ett givet syfte och för en avsedd användare eller uppdragsgivare.

Eleven nyttjar med **god säkerhet** datautbyte med datakällor och identifierar, åtgärdar och förhindrar fel vid programmering och exekvering på ett **väl fungerande** sätt.

Betyget B

Elevens kunskaper bedöms sammantaget vara mellan A och C.

Betyget A

Eleven för **välutvecklade och nyanserade** resonemang om egenskaper hos samt för- och nackdelar med olika programmeringsparadigm och programspråk.

Eleven väljer och använder med **mycket god säkerhet** designmönster, algoritmer och datastrukturer.

Eleven skapar **ändamålsenliga och effektiva** program med ett givet syfte och för en avsedd användare eller uppdragsgivare.

Skolverket

Vidareutbildning i form av ett fjärde tekniskt år i gymnasieskolan

Eleven nyttjar med **mycket god säkerhet** datautbyte med datakällor och identifierar, åtgärdar och förhindrar fel vid programmering och exekvering på ett **mycket väl fungerande** sätt.

FÖRSLAG