

Programmering C – valgfag, august 2017

1. Identitet og formål

1.1. Identitet

Programmering omfatter de metoder og teknikker, der anvendes til at få programmerbare it-komponenter til at udføre planlagte handlinger. Faget har en betydelig praktisk dimension med programmering af enkle it-komponenter og demonstration af mere komplekse teknologiske systemer. Faget er et teknisk og kreativt fag med vægt på eksperimentelle og innovative processer, samt et videns- og kundskabsfag. Disse sider af faget betinger gensidigt hinanden og sikrer faglig dybde.

1.2. Formål

Formålet med faget er at give en bred og nuanceret introduktion til programmering, herunder at illustrere brug af programmering med henblik på at tilfredsstille personlig nysgerrighed, til kreativt udtryk, til at skabe ny viden og til at løse problemer (til hjælp for mennesker, organisationer og samfund).

Faget bidrager til elevernes almindelse og studiekompetence ved at udvikle evnen til logisk, systematisk og kreativ tankegang og ved at udvikle særlige programmeringsorienterede it-færdigheder, der kan anvendes inden for andre fagområder i uddannelsen og i videreuddannelsessammenhæng. Gennem arbejdet med faget opnår eleverne kompetence til at kunne undersøge og beskrive enkle processer og behandle data og informationer ved hjælp af programmering.

Faget understøtter elevernes muligheder for at agere i den globale højteknologiske verden. Faget giver eleverne viden om, hvordan it-systemer udvikles, og om it-systemers potentialer for løsninger af globale, nationale og lokale problemstillinger.

2. Faglige mål og fagligt indhold

2.1. Faglige mål

Eleverne skal kunne:

- læse enkle programmer og redegøre for deres funktionsmåde og anvendelsesmuligheder
- rette og tilpasse enkle programmer
- anvende eksisterende programdele og biblioteksmoduler i arbejdet med at programmere et fungerende system
- demonstrere systematik i programmeringsprocessen
- løse en enkel problemstilling gennem udviklingen af et program bl.a. i samspil med andre fag
- anvende grundlæggende konstruktioner i et programmeringssprog
- demonstrere viden om fagets identitet og metoder.

2.2. Kernestof

Gennem kernestoffet skal eleverne opnå faglig fordybelse, viden og kundskaber.

Kernestoffet er:

- programmeringssprog og elementer i programmeringssprogets opbygning såsom data- og kontrolstrukturer
- programdele og biblioteksmoduler
- arbejdsgange og systematik i programmeringsprocessen
- abstrakte programmeringsbeskrivelser og dokumentation.

2.3. Supplerende stof

Eleverne vil ikke kunne opfylde de faglige mål alene ved hjælp af kernestoffet.

I forhold til de faglige samspil med de øvrige fag i uddannelsen vælges der supplerende stof med henblik på at bibringe faglig fordybelse og styrke toningen af kernestoffet. Dele af det supplerende stof vælges i samarbejde med eleverne, når det er muligt.

Der kan indgå læsning af it-faglige tekster på engelsk samt, når det er muligt, på andre fremmedsprog.

2.4. Omfang

Det forventede omfang af fagligt stof er normalt svarende til 90-150 sider.

3. Tilrettelæggelse

3.1. Didaktiske principper

Undervisningen organiseres omkring flere temaer og projekter. Disse vælges, så de tilsammen dækker kernestof og supplerende stof, med henblik på at der er en klar progression i arbejdet med faglige mål og selvstændighed.

Undervisningen tilrettelægges ved brug af anerkendte didaktiske principper, herunder 'use-modify-create'-progression fra at anvende udleverede programmer til at modificere disse for til sidst selvstændigt at skabe (nye dele af) programmer; 'Stepwise Improvement', som teknik til trinvis, iterativ og systematisk udvikling af programmer og 'Worked Examples' (kombineret med 'faded guidance'), til illustration af eksemplariske løsningsprocesser.

Undervisningsformen differentieres således, at alle elever udvikler sig i undervisningsforløbet. Der veksles mellem overbliksskabende forløb, eksperimenter, øvelser og projekter.

3.2. Arbejdsformer

Undervisningen tager udgangspunkt i elevens hverdag og teknologier, der indeholder programmerede funktioner. Der lægges vægt på, at eleven kan beskrive programmets funktion i normalt sprog og opnå en naturlig tilgangsvinkel til at omsætte disse funktioner til elementer i et programmeringssprog.

Der vælges et primært programmeringssprog som grundlag for undervisningen. Der arbejdes med eksempler på programmer. Udviklingen af forståelsen af programmeringssproget sker ved at eksperimentere med varianter af enkle programmer. Arbejdet med aktuelle teknologier skal sætte eleverne i stand til at reflektere over egne evner og interesse for karriere indenfor programmering eller andre fagområder, hvor programmeringskompetencen er relevant.

Undervisningen tilrettelægges om muligt med udadrettede aktiviteter og/eller i samarbejde med eksterne parter, som eksemplificerer fagets anvendelses- og karrieremuligheder.

Undervisningsformen fremmer en progression i både indholdsmæssig sværhedsgrad og selvstændighed i problemløsningen. Der udarbejdes produkter med tilhørende dokumentation i form af synopsis, herunder et eksamensprojekt.

I den afsluttende periode af undervisningen afsættes 20 timers undervisningstid til, at eleverne med vejledning fra læreren udarbejder et eksamensprojekt i grupper på to til tre. Hvor dette ikke er muligt eller ønskeligt, kan man lade eleverne arbejde individuelt. Eksamensprojektet består af et produkt og en synopsis. Synopsen skal dokumentere udviklingen af det færdige produkt og har et omfang på fem til otte normalsider, eksklusiv koder, rutediagrammer, bilag mm.

Eksamensprojektet udarbejdes inden for rammerne af et projektoplæg stillet af skolen. Eksamensprojektgrupperne udarbejder en fælles projektbeskrivelse, der inkluderer en beskrivelse af den enkelte eksaminands fokus. Projektbeskrivelsen godkendes af skolen, når beskrivelsen er tilstrækkelig fagligt bred og niveaumæssigt relevant.

Eksamensprojektets synopsis er individuelt udarbejdet.

Afleveringstidspunktet skal normalt være senest en uge før eksamensperiodens begyndelse.

Eksamensprojektet indgår i grundlaget for den afsluttende standpunktskarakter, hvis der gives en sådan, og udgør grundlaget for prøven.

Eksamensprojektets synopsis er forinden prøven ikke rettet og kommenteret af eksaminator.

Den enkelte elev dokumenterer løbende sin faglige udvikling i en logbog. Dokumentationen i logbogen kan have form af f.eks. programmer, noter, synopsis, programbeskrivelser og flowcharts.

3.3. It

Gennem arbejdet med udvikling af programmer i faget opnås såvel specifikke faglige digitale kompetencer som almene digitale kompetencer, hvilket er fagets bidrag til uddannelsernes overordnede krav om digital dannelse.

- Programmeringsværktøjer, der automatisk kan generere dokumentation og debugging, anvendes, ligesom andre informationsteknologiske værktøjer inddrages efter behov.
- Internettet anvendes som søgningsværktøj til oplysninger, vejledninger, eksempler, programdele og biblioteksmoduler med efterlevelse af ophavsretlige regler og dokumentationskrav.
- I arbejdet med stof om konkrete teknologier og standardiseringer skal eleverne anvende originale kilder (eksempelvis dokumentation af programmeringssprog, data og diagrammer).
- Eleverne arbejder med digital dokumentation af deres programmer, bl.a. i form af modeller og kommentarer i programmeringskoden.

3.4. Samspil med andre fag

Dele af kernestof og supplerende stof skal vælges og behandles, så det kan bidrage til styrkelse af det faglige samspil mellem fagene og i studieretningen. I tilrettelæggelsen af undervisningen inddrages elevernes viden og kompetencer fra andre fag, som eleverne hver især har, så de bidrager til perspektivering af temaer og belysning af fagets almindelige sider.

Når faget indgår i flerfaglige forløb, lægges der vægt på, at eleverne får mulighed for løbende at reflektere over, hvordan deres valg og behandling af viden og metoder fra de indgående fag påvirker kvaliteten af den flerfaglige problemløsning.

Der lægges i undervisningen vægt på, at programmering er en aktivitet, der kan hjælpe med at løse opgaver fra andre fagområder. Der kan inddrages problemstillinger fra elevernes øvrige fag, og eleverne kan få mulighed for at bruge programmering til at belyse og løse problemer på nye måder og derved skabe ny indsigt gennem programmering. Eksamensprojektet kan udformes som en del af et projekt i et andet fag eller ved at inddrage viden fra andre fag.

4. Evaluering

4.1. Løbende evaluering

Eleverne udarbejder i undervisningsperioden et antal produkter i form af programmer med tilhørende dokumentation. Eleven samler noter, produkter og dokumentationer i sin logbog, som anvendes i forbindelse med elevens selvevaluering og den fremadrettede evaluering ved samtaler med og feedback fra læreren. Evalueringen skal give en individuel vurdering af niveauet for - og udviklingen af elevens faglige standpunkt i forhold til opfyldelse af de faglige mål.

4.2. Prøveform

Der afholdes en mundtlig individuel prøve på grundlag af eksaminandens eksamensprojekt, jf. pkt. 3.2.

Før prøven sender skolen et eksemplar af synopserne, samt en fortegnelse over eksamensprojekt- beskrivelserne, til censor. Eksaminator og censor drøfter inden prøven, hvilke problemstillinger i eksamensprojektet eksaminanden skal uddybe. Eksaminationstiden er ca. 24 minutter. Der gives ingen forberedelsestid. Eksaminationen tager udgangspunkt i eksaminandens præsentation af eksamensprojektet suppleret med et eller flere i forvejen forberedte spørgsmål fra eksaminator. Eksaminationen former sig derefter som en samtale mellem eksaminand og eksaminator med udgangspunkt i synopsen.

4.3. Bedømmelseskriterier

Bedømmelsen er en vurdering af, i hvilken grad eksaminandens præstation opfylder de faglige mål, som de er angivet i pkt. 2.1.

Ved prøve, hvor faget indgår i samspil med andre fag, lægges der vægt på, at eksaminanden kan

- demonstrere viden om fagets identitet og metoder
- behandle problemstillinger i samspil med andre fag.

Der gives én karakter ud fra en helhedsvurdering af eksaminandens mundtlige præstation.