



Programmeringsteknik Computer programming

7,5 högskolepoäng

7,5 credits

Ladokkod: NPT021

Version: 1.0

Fastställd av: Institutionsstyrelsen 2008-06-19

Gäller från: HT 2008

Nivå: Grundnivå

Huvudområde (successiv fördjupning): Informatik (G1N)

Utbildningsområde: Naturvetenskap

Ämnesgrupp: Informatik/Data- och systemvetenskap

Förkunskapskrav: Grundläggande behörighet + Matematik 3b / 3c, Samhällskunskap 1b / 1a1 +1a2.

Eller:

Engelska B, Matematik C, Samhällskunskap A. Grundläggande behörighet samt lägst betyg G/3 i Ma C och Sh A (Områdesbehörighet 4)

Betygsskala: Underkänd, Godkänd eller Väl godkänd

Innehåll

Kursen är en grundkurs i programmering. Kursens behandlar därför dels generella grundläggande koncept vid strukturerad programmering och dels programspråket C. Kursen varvar teori med praktik på så sätt att varje huvudavsnitt först behandlas vid föreläsningar och övningar för att därefter övas och prövas på laborationerna. Under laborationsperioderna arbetar studenterna enbart med respektive laboration; dvs. då hålls inga föreläsningar eller övningar. De moment som ingår i kursen är

- **Grundläggande koncept i programmering och programmeringsspråk.** Här behandlas t.ex. variabler, operatorer och datatyper.
- **Kontrollstrukturer i C.** Kursen innehåller en genomgång av samtliga grundläggande kontrollstrukturer dvs. olika villkorssatser och konstruktioner för att uppnå iteration (loopar).
- **Funktioner.** Principer för hur ett program bör delas upp i funktioner går igenom. Den parameteröverföringsteknik som konsekvent används i C (call-by-value) penetreras i detalj.
- **Stränghantering.** Kursen använder ett kodbibliotek som förenklar stränghantering. Med detta bibliotek kan strängar behandlas antingen som en enhet eller som en samling (vektor av) tecken.
- **Modularisering.** Ett mycket viktigt inslag i all programdesign är uppdelning i entiteter. I denna kurs tillämpas principen att funktionalitet och representation bör skiljas åt. Rent konkret åstadkoms detta genom att skapa moduler där funktionaliteten beskrivs i gränssnittet medan representationen läggs separat i en annan fil.
- **Pekare.** Grundläggande pekaroperationer och pekararitetik går igenom. Framförallt fokuseras på nyttan med pekare; dvs. i denna kurs, främst hur pekare används för att uppnå call-by-reference, hur pekare möjliggör delande av stora datastrukturer samt hur pekare används vid dynamisk minneshantering.
- **Dynamisk minneshantering.** I kursen används normalt speciella bibliotek för att förenkla hanteringen av dynamiskt minne, men de mer primitiva kommandon som finns tillgängliga i standard C går också igenom.
- **Vektorer.** Olika idiom för hantering av vektorer behandlas. Dessutom behandlas hur vektorer skickas som argument till funktioner och hur de returneras från funktioner.
- **Poster.** Principen för att skapa en post i C är att man först introducerar en egendefinierad typ och därefter skapar variabler av den nya typen. I kursen beskrivs hur detta genomförs samt för- och nackdelar med att låta den nya typen vara en pekartyp. Överförandet av poster från och till funktioner, samt hantering av dynamiskt minne i samband med poster avhandlas också.
- **Sökning och sortering.** De generiska uppgifterna sökning och sortering introduceras. Basala sök- och sorteringsalgoritmer behandlas.
- **Filhantering.** Grundläggande principer och konstruktioner i C för skapande, öppnande, skrivande och läsande av textfiler behandlas.
- **Felsökning.** Enklare tekniker för att felsöka egna program under utveckling presenteras.

- **Kodkonventioner.** För att skapa program som är lätta att underhålla och utöka bör kodkonventioner följas. Dessa konventioner styr t. ex. namngivning av variabler och funktioner samt val av idiom. En konkret detalj är att namngivna konstanter skall användas istället för numeriska värden i koden.
- **Koddokumentation.** Nyttan av väl kommenterade program belyses. I kursen används kommentarer på programnivå, på funktionsnivå och på satsnivå. Programkommentarer förklarar programmets uppgift och användning, samt ger en översiktlig uppdelning. Funktionskommentarer förklarar en funktions uppgift och framförallt hur den skall användas; dvs. dess input och output. Satskommentarer förklarar en enskild sats, där så är nödvändigt för att programmets exekvering skall gå att följa.
- **Principer vid programkonstruktion.** Kursen använder genomgående ett top-down perspektiv vid programkonstruktion. Mer konkret bryts ett större problem ned i successivt mindre problem. Två viktiga principer i detta arbetssätt är att strukturen hela tiden skall motsvaras av programkoden och att programmet skall byggas för att möjliggöra kontinuerlig testning.

Mål

Kursens mål är att ge en teoretisk och praktisk grund för programmering av enklare applikationer. Studenten ska kunna tillämpa samtliga ingående teoretiska moment (se under innehåll) och därmed specifikt kunna

- självständigt konstruera enkla applikationer med textbaserat gränssnitt utifrån en given kravspecifikation.
- redogöra för, motivera och tillämpa de viktigaste kriterierna för vad som utmärker god programvara.
- analysera och stegvis bryta ner ett problem till en nivå som möjliggör kodning i C.
- tillämpa en strukturerad ansats för programkonstruktion.

Efter kursen ska studenten kunna

- Redogöra för och använda de olika inbyggda datatyperna i språket C.
- Använda de grundläggande konstruktionerna i språket C; dvs. satser, operatorer, villkor och iteration.
- Använda de sammansatta datatyperna vektorer och poster.
- Redogöra för konceptet pekare samt kunna använda pekare då så krävs.
- Redogöra för och tillämpa den princip för parameteröverföring som gäller i C.
- Tillverka välstrukturerade program genom att välja en lämplig uppdelning i funktioner och moduler.
- Välja en programdesign som uppfyller de viktigaste kriterierna för god programvara.
- Systematiskt felsöka i ett egenutvecklat program.
- Använda och anpassa kodbibliotek.
- Redogöra för och tillämpa den generiska programmeringsmetodiken ”stegvis nedbrytning” (stepwise refinement).

Undervisningsformer

Föreläsningar, övningar och laborationer med tillhörande handledning. Föreläsningarna introducerar och exemplifierar den viktigaste teorin. Övningarna innebär att studenter i samarbete med lärare löser övningar, antingen på papper eller vid dator. I laborationerna löser studenterna självständigt relativt komplexa uppgifter, vilka normalt innebär att från en kravspecifikation skriva ett program. Undervisningen bedrivs normalt på svenska, men undervisning på engelska kan förekomma.

Examinationsformer

Examination på kursen består av tre obligatoriska laborationer och en tentamen. Tentamen genomförs som en kombinerad enskild prövning, innehållande en skriftlig, teoretisk del och en programmeringsdel vid dator. Laborationerna betygssätts som antingen godkänt eller underkänt. På tentamen ges ett av betygen väl godkänt, godkänt eller underkänt.

Laborationsmomentet ger 2,5 högskolepoäng.

Tentamen ger 5,0 högskolepoäng.

Kursens färdighetsmål examineras främst genom laborationerna och datortentamen. Programmeringsdelen av tentamen prövar särskilt en enskild students förmåga att koda mindre program utifrån en problembeskrivning. Normalt fokuserar varje uppgift på programmeringsdelen ett principiellt viktigt koncept.

Kunskapsmålen examineras i första hand i den skriftliga delen av kursens tentamen.

För att erhålla betyget Godkänd på hel kurs krävs godkänt betyg på var och en av laborationerna samt godkänt på tentamen. För betyget Väl godkänt krävs dessutom väl godkänt på tentamen.

Studentens rättigheter och skyldigheter vid examination är enligt riktlinjer och regelverk vid Högskolan i Borås.

Studentens rättigheter och skyldigheter vid examination är enligt riktlinjer och regelverk vid Högskolan i Borås.

Kurslitteratur och övriga läromedel

Roberts, Eric.S, The art and science of C, ISBN: 0-201-54322-2

Studentinflytande och utvärdering

Kursutvärderingen är skriftlig. Utvärderingen sammanställs och offentliggörs i enlighet med institutionens bestämmelser. Sammanställningen samt eventuella planerade förändringar till nästa kursgenomförande delges studenterna.

Övrigt

Kursen ingår i Systemarkitekturutbildningen. Kursen ersätter kursen Programmeringsteknik (NPT011). Kursen ges även som fristående kurs.