



Programmeringsteknik Computer programming

7,5 högskolepoäng

7,5 credits

Ladokkod: NPT021

Version: 3.1

Fastställd av: Utskottet för utbildningar inom bibliotek, information och IT 2018-05-29

Gäller från: HT 2018

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. Kursen behandlar därför dels generella grundläggande koncept vid strukturerad programmering och dels programspråket C. De moment som ingår i kursen är:

- **Grundläggande koncept i programmering och programmeringsspråk.** Här behandlas t.ex. variabler, operatörer och datatyper.
- **Kontrollstrukturer i C.** Kursen innehåller en genomgång av samtliga grundläggande kontrollstrukturer, dvs olika villkorssatser och konstruktioner för att uppnå iterationer (loopar).
- **Funktioner.** Principer för hur ett program bör delas upp i funktioner går igenom. Den parameteröverföringsteknik som konsekvent används i C (call-by-value) presenteras i detalj.
- **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.
- **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.
- **Vektorer (arrayer).** Olika idiom för hantering av vektorer (arrayer) behandlas. Dessutom behandlas hur vektorer (arrayer) skickas som argument till funktioner och hur de returneras från funktioner.
- **Sökning och sortering.** De generiska uppgifterna sökning och sortering introduceras. Basala sök- och sorteringsalgoritmer behandlas.
- **Pekare.** Grundläggande pekaroperationer och pekararitmetik går igenom. Framförallt fokuseras på nyttan med pekare: dvs 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.
- **Poster (structs).** Principen för att skapa en post i C är att man först introducerar en egendefinerad 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 (structs) från och till funktioner, samt hantering av dynamiskt minne i samband med poster (structs) avhandlas också.
- **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

Efter avslutad kurs ska studenten kunna, med avseende på,

Kunskap och förståelse

- 1.1 redogöra för, och motivera, de viktigaste kriterierna för vad som utmärker god programvara,
- 1.2 redogöra för kompilers- och länkningsprocessen för ett C-program,
- 1.3 redogöra för de inbyggda datatyperna och kontrollstrukturerna i språket C,
- 1.4 redogöra för funktioner och principen för parameteröverföring i språket C,
- 1.5 redogöra för modularisering av ett program inklusive interface och representation,
- 1.6 redogöra för stränghantering i språket C,
- 1.7 redogöra för de sammansatta datatyperna vektorer (arrays) och poster (structs) i språket C,
- 1.8 redogöra för konceptet pekare och dynamisk minneshantering,
- 1.9 redogöra för filhantering i språket C,
- 1.10 redogöra för den generiska programmeringsmetodiken för stegvis nedbrytning (stepwise refinement).

Färdighet och förmåga

- 2.1 utifrån en kravspecifikation kunna konstruera enkla applikationer med textbaserat gränssnitt,
- 2.2 analysera och stegvist bryta ner ett problem till en nivå som möjliggör implementering i språket C,
- 2.3 tillämpa vedertagna krav på kodkonventioner, dokumentation och principer för god programvarukonstruktion,
- 2.4 tillämpa de inbyggda datatyperna och kontrollstrukturerna i språket C,
- 2.5 tillämpa funktioner och principen för parameteröverföring i språket C,
- 2.6 tillämpa modularisering samt separation av interface och representation,
- 2.7 tillämpa stränghantering i språket C,
- 2.8 tillämpa vektorer (arrayer) i språket C,
- 2.9 tillämpa poster (structs) i språket C,
- 2.10 tillämpa pekare och dynamisk minneshantering på ett korrekt sätt,
- 2.11 tillämpa filhantering i språket C,
- 2.12 tillämpa den generiska programmeringsmetodiken för stegvis nedbrytning (stepwise refinement),
- 2.13 tillämpa och anpassa kodbibliotek,
- 2.14 systematiskt felsöka i ett egenutvecklat program.

Värderingsförmåga och förhållningssätt

- 3.1 uppvisa ett strukturerat förhållningssätt till programvarukonstruktion utifrån ett givet problem.
- 3.2 kritiskt granska och utvärdera kod med avseende på vedertagna konventioner och principer för god programvarukonstruktion.

Undervisningsformer

Undervisningen på kursen består av

- föreläsningar
- handledning i workshopsform
- handledning av laborationer

Föreläsningarna presenterar kursens teoretiska innehåll. Handledning sker för grupplaborationer, vilka utgör kursens praktiska innehåll. På varje workshop arbetar studenterna, under överinseende av lärare, med en större uppgift som sedan går igenom och diskuteras gemensamt.

Undervisningen bedrivs på svenska, men undervisning på engelska kan förekomma.

Examinationsformer

Kursen examineras genom grupplaborationer samt en individuell salstentamen:

Tentamen: individuell salstentamen

Lärandemål: 1.1 - 1.10, 2.1 - 2.14, 3.1 - 3.2

Högskolepoäng: 5,0

Betygsskala: UVG

Laboration 1: kontrollstrukturer och funktioner (gruppuppgift)

Lärandemål: 2.1 - 2.5, 2.12 - 2.14

Högskolepoäng: 0,5

Betygsskala: UG

Laboration 2: moduler, strängar och vektorer (gruppuppgift)

Lärandemål: 2.1 - 2.8, 2.12 - 2.14

Högskolepoäng: 1,0

Betygsskala: UG

Laboration 3: pekare, minneshantering och filhantering, mm (gruppuppgift)

Lärandemål: 2.1 - 2.14

Högskolepoäng: 1,0

Betygsskala: UG

För betyget Godkänd på hel kurs krävs Godkänd på samtliga laborationer och tentamen. För betyget Väl Godkänd på hel kurs krävs dessutom Väl Godkänd på tentamen.

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

Kurslitteratur och övriga läromedel

Kurslitteraturen är på engelska.

Roberts, E. S. (1994), The Art and Science of C: A Library Based Introduction to Computer Science, Pearson, ISBN: 0-201-54322-2

Vetenskapliga artiklar och föreläsningmaterial kan tillkomma enligt lärarens anvisningar.

Studentinflytande och utvärdering

Kursen utvärderas i enlighet med gällande riktlinjer för kursvärderingar vid Högskolan i Borås, där studenternas synpunkter ska inhämtas. Kursutvärderingsrapporten publiceras och återkopplas till deltagande och blivande studenter i enlighet med ovan nämnda riktlinjer, och ligger till grund för framtida utveckling av kurser och utbildningsprogram. Kursansvarig lärare ansvarar för att utvärdering enligt ovan genomförs.

Övrigt

Kursen ingår i Systemarkitekturutbildningen.