



Data- och programstrukturer Programming Abstractions and Data Structures

7,5 högskolepoäng

7,5 credits

Ladokkod: NDP011

Version: 4.1

Fastställd av: Utskottet för utbildningar inom bibliotek, information och IT 2015-12-15

Gäller från: VT 2016

Nivå: Grundnivå

Huvudområde (successiv fördjupning): Datalogi (G1F), Informatik (G1F)

Utbildningsområde: Naturvetenskap

Ämnesgrupp: Informatik/Data- och systemvetenskap

Förkunskapskrav: Avklarad kurs Programmeringsteknik 7,5 högskolepoäng eller motsvarande.

Betygsskala: Underkänd, Godkänd eller Väl godkänd

Innehåll

Kursen är en fortsättningskurs i strukturerad programmering. Kursen behandlar därför dels generella begrepp och generella lösningar vid strukturerad programmering och dels mer avancerade koncept i programspråket C.

Kursen behandlar

- Rekursion: Rekursiva algoritmer används flitigt genom hela kursen varför det rekursiva tankesättet introduceras tidigt och i detalj. Typiska algoritmer som behandlas är MiniMax och olika varianter på backtracking.
- Abstrakta datatyper (ADT:er): teoretisk motivation, principer för design av ADT:er, val av konkret representation och implementering i C. De viktigaste ADT:erna som går igenom i kursen är: stack, kö, lista, scanner, textbuffert, symboltabell. Användning av ADT:er för att skapa avancerade applikationer, exempelvis texteditorer.
- Komplexitet: Resonemang om komplexitet används för att analysera olika implementationer av (främst) ADT:er.
- Linjära datastrukturer: Stackar, köer, prioritetsköer och länkade listor, samt representationer som kombinerar länkade listor och vektorer. Även hashning och hashtabeller behandlas. Särskilt fokus läggs på hur intern representation påverkar olika operationers effektivitet för respektive datastruktur.
- Rekursiva datastrukturer och algoritmer för att hantera dessa: listor och träd, med särskilt fokus på binära sökträd och algoritmer för insättning, borttagning och sökning efter noder och traversering av träd.
- Uttrycksträd och interpretatorer: Samtliga steg från språkspecifikation till evaluering går igenom för ett enkelt formellt språk.
- Avancerade konstruktioner i språket C och tillämpningar av dessa: pekare och dynamisk minnesallokering för att skapa olika länkade strukturer, funktionspekare och unioner.

Praktisk erfarenhet ges via laborativa projekt där studenter i grupp använder och implementerar datastrukturer för att bygga applikationer i C.

Mål

Efter avklarad kurs ska studenten kunna, med avseende på,

Kunskap och förståelse

- 1.1 redogöra för och motivera nyttan av koncepten ”inkapsling”, ”abstrakta datatyper” och ”programmering mot gränssnitt”,
- 1.2 beskriva de datastrukturer och abstrakta datatyper som kursen behandlar,
- 1.3 beskriva algoritmer för hantering av de datatyper som kursen behandlar,
- 1.4 utifrån teorin redogöra för och motivera val av konkret implementation för de abstrakta datatyper som kursen behandlar,
- 1.5 redogöra för avancerade konstruktioner i språket C,

Färdighet och förmåga

- 2.1 självständigt kunna konstruera avancerade applikationer med textbaserat gränssnitt utifrån en given kravspecifikation,
- 2.2 utnyttja abstrakta datatyper för att erhålla en god programdesign,
- 2.3 såväl utnyttja som implementera rekursiva algoritmer och rekursiva datatyper,
- 2.4 givet en enkel grammatik och semantik kunna konstruera en interpretator för uttryck i det specificerade språket,
- 2.5 givet en kravspecifikation kunna designa ett gränssnitt för en abstrakt datatyp,
- 2.6 skapa inkapslade datatyper i språket C,
- 2.7 genomföra enklare komplexitetsanalys av iterativa algoritmer,
- 2.8 använda och implementera de datastrukturer och abstrakta datatyper som kursen behandlar,
- 2.9 använda och implementera algoritmer för hantering av de datatyper som kursen behandlar och
- 2.10 använda avancerade konstruktioner i språket C,

Värderingsförmåga och förhållningssätt

- 3.1 välja och utnyttja lämpliga abstrakta datatyper vid programkonstruktion,
- 3.2 välja en lämplig konkret representation för en given abstrakt datatyp.

Undervisningsformer

Undervisningen består av föreläsningar och laborationer.

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

Examinationsformer

Tentamen: (TE11)

Lärandemål: 1.1-1.5, 2.3-2.10, 3.1-3.2

Högskolepoäng: 5.0

Betygskala: UVG

Laboration 1 (LA11): Programmeringsuppgift i grupp

Lärandemål: 2.3, 2.10

Högskolepoäng: 0.5

Betygskala: UG

Laboration 2 (LA12): Programmeringsuppgift i grupp

Lärandemål: 2.3, 2.6-2.10

Högskolepoäng: 0.5

Betygskala: UG

Laboration 3 (LA13): Programmeringsuppgift i grupp

Lärandemål: 2.1-2.6, 2.8-2.10, 3.1-3.2

Högskolepoäng: 1.5

Betygskala: UG

För betyget Godkänd på hel kurs krävs minst godkänt betyg på samtliga moment. För betyget Väl godkänd krävs dessutom Väl Godkänt på tentamen.

Då kursplanen ändras kommer student som önskar slutföra rester från ett kurstillfälle att examineras utifrån kursens nya innehåll och upplägg. Då kursen har upphört kan student som önskar slutföra rester följa hela eller delar av annan likvärdig kurs.

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

Kurslitteratur och övriga läromedel

Litteratur är i huvudsak på engelska men även svenska kan förekomma.

Roberts, E. S. Programming Abstractions in C – A Second Course in Computer Science, Addison-Wesley, 1998.

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 ges inom Systemarkitekturutbildningen.