

## Kursbeskrivning för kursen Programmering (kurskod DT2012)

*Välkommen till kursen Programmering!* Detta blad innehåller grundläggande information som gäller för kursen Programmering (7.5 hp, kurskod DT2012) som ges av Högskolan i Halmstad under höstterminen 2016.

### Hemsida

Kursens löpande information, inklusive föreläsningsanteckningar, övningsuppgifter, inlämningsuppgifter samt nyheter och länkar till bland annat schema kommer att finnas under

[bit.ly/2bKGirx](http://bit.ly/2bKGirx)

som du även når när du loggar in i Blackboard genom studentportalen. *Du kan behöva använda länken två gånger. Sidan finns men Blackboard kan säga att sidan inte finns första gången du försöker komma åt den.*

### Lärare

Kursansvarig och examinator är Verónica Gaspes. Du kan alltid komma i kontakt med henne via e-post

[veronica.gaspes@hh.se](mailto:veronica.gaspes@hh.se)

Ange alltid i ämnesraden att det handlar om kursen programmering så att brevet sorteras rätt i hennes inkorg.

I kursen kommer även Nicolina Månsson (lärare) och några studenter att delta som handledare och övningsledare.

## Kurslitteratur och annat material

Läraren kommer att följa boken *Introduction to Programming in Java. An interdisciplinary approach* av Sedgewick och Wayne. Boken är mycket bra och har en omfattande webbplats med användbart material

[www.cs.princeton.edu/introcs](http://www.cs.princeton.edu/introcs)

Bland annat finns hela kapitel 1 tillgängligt som pdf:

[introcs.cs.princeton.edu/java/home/chapter1.pdf](http://introcs.cs.princeton.edu/java/home/chapter1.pdf)

Dessutom kommer du att använda boken i en senare kurs om algoritmer och datastrukturer.

Vi rekommenderar att du ser till att ha i din dator den mjukvara vi kommer att använda i kursen. I kursboken finns instruktioner för att ladda ner Javas programmeringsverktyg för

[introcs.cs.princeton.edu/java/mac/](http://introcs.cs.princeton.edu/java/mac/)  
[introcs.cs.princeton.edu/java/linux/](http://introcs.cs.princeton.edu/java/linux/)  
[introcs.cs.princeton.edu/java/windows/](http://introcs.cs.princeton.edu/java/windows/)

Det finns mycket material på nätet för att lära sig programmering med Java. Till exempel finns det en bra online bok *Introduction to Programming using Java, 7th edition* av David Eck,

[math.hws.edu/javanotes/](http://math.hws.edu/javanotes/)

och en bra introduktionskurs som ges av Cay Horstmann i Udacity *Introduction to Java Programming*

<https://www.udacity.com/course/intro-to-java-programming--cs046>

Kurser på Udacity betalar man för (du ska INTE göra det!), men de har mycket material som man inte betalar för (kolla *instructor videos*). Jag kommer att länka från kursens hemsida till några korta videos när de handlar om samma saker som vi behandlar i vår kurs.

## Syfte

Kursens syfte är att ge grundläggande kunskaper och färdigheter i att programmera på ett systematiskt sätt i ett objektorienterat programspråk, till exempel Java. Kursen är förberedande för fortsatta studier i ämnet.

Det finns två saker som folk brukar vilja få ut av en kurs i programmering:

1. Att tänka som en problemlösare; till att börja med, att tänka *algoritmiskt*. Är du intresserad av detta kan du redan nu kolla Computational Thinking, en kort artikel av Jeannette Wing under

<http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf>

2. Att behärska ett programspråk; i vårt fall Java. Är du intresserad kan du redan nu kolla

<http://introc.cs.princeton.edu/java/home/>

webplatsen för den bok vi kommer att använda i kursen.

I en programmeringskurs brukar man försöka få in båda delarna! Detta medför stora risker att några studenter gillar vissa delar mer än andra. Men vi har stora förhoppningar om att många fastnar för den fantastiska känslan av att både lösa problem och tillverka saker som fungerar!

## Lärandemål

Efter avslutad kurs skall du kunna

### *Kunskap och förståelse*

- analysera och förklara beteendet av enkla program

Med detta menas att du skall kunna läsa enkla program skrivna av dig eller andra och förklara vad programmet gör när det används. Till exempel skall du kunna förklara hur många gånger en viss instruktion upprepas, i vilken ordning instruktionerna utförs, vilka värden variabler i programmet får, vilka metoder som anropas, vilken input programmet tar och vilken output programmet genererar.

Detta examineras i duggor och den skriftliga tentamen.

- redogöra för begrepp inom imperativ objektorienterat programmering som de förekommer i det programspråk som används i kursen.

Med detta menas att du till exempel skall kunna redogöra för skillnaden mellan primitiva typer och referenstyper samt skillnaden mellan uttryck och kommandon. Du skall kunna redogöra för vad tilldelning är och för de styrstrukturer som kan användas för att upprepa kommandon (for- och while-loop), för att välja mellan kommandon (if-then-else och switch). Du skall kunna redogöra för fält och strängar. Du skall kunna redogöra för klasser, objekt, konstruerare och metoder. Du skall kunna redogöra för argument och kommandoradsargument.

Detta examineras i duggor och den skriftliga tentamen.

### *Färdighet och förmåga*

- konstruera enkla algoritmer med hjälp av programspråkets styrstrukturer och datatyper

Med detta menas att du skall själv kunna, efter en tydlig beskrivning av ett problem, välja vilka datatyper och vilka kommandon programmet skall använda sig av för att lösa problemet. Du måste även kunna skriva programmet i Java, kompilera programmet och använda det. Du skall kunna hantera variabler, heltal, flyttal, fält och strängar. Du skall kunna använda tilldelning, loopar, if- och switch-satser samt metदानrop.

Detta examineras i projekt och den skriftliga tentamen.

- organisera program i moduler med hjälp av klasser som kapslar in data och som har metoder som gränssnitt

Med detta menas att du skall, efter tydliga anvisningar, kunna definiera en klass med tillståndsvariabler, konstruerare och metoder som kan fungera som del i program (alltså som moduler).

Detta examineras i projekt.

- använda programbibliotek för in- och utmatning, grafik och användargränssnitt.

Med detta menas att du skall kunna, i dina program, använda program som andra har skrivit och som finns tillgängliga. Under kursen kommer vi att visa hur man använder sådana för att kunna ta in data på olika sätt, för att generera utdata på olika sätt, för att rita och för att interagera med användaren med traditionella användargränssnitt så som knappar, menyer, etc.

Detta examineras i projekt och den skriftliga tentamen.

#### *Värderingsförmåga och förhållningssätt*

- identifiera värdet av programmering som verktyg i andra ämnen, i det fortsatta karriären samt behov av ytterligare kunskaper inom ämnet.

Med detta menas att du skall kunna använda programmering i andra ämnen eller åtminstone se möjligheter till detta. Det kan handla om att visualisera stora mängder data, eller lösa problem som måste hantera stora mängder data eller som kräver svåra eller många beräkningar. Det kan handla om ämnen som matematik, signalbehandling, reglerteknik, artificiell intelligens, med mera.

Detta examineras i duggor och projekt.

## Undervisning och examination

Undervisningen är organiserad som en blandning av föreläsningar och efter-läsningar (vi kommer att prata mer om vad detta kommer att innebära under första kurs-tillfälle), övningsdiskussioner samt projekthandledning. Första veckan är speciellt, med två föreläsningar och ett övningstillfälle, annars kommer vi att arbeta enligt följande:

- Varannan vecka:
  - en föreläsning (två timmar, hela klassen),
  - en övningsdiskussion (två timmar, 12 studenter) och
  - en projekthandledning (två timmar, 24 studenter)
  
- Varannan vecka:
  - en efter-läsning (med dugga, 3 timmar, hela klassen) och
  - en projekthandledning (två timmar, 24 studenter)

För att få godkänt på kursen måste du få godkänt på sex av åtta duggor, på fyra projekt och på en skriftlig tentamen. Alla projekt lämnas in i Blackboard-systemet enligt de deadlines som anges i projektbeskrivningen och du kommer att få besked senast en vecka efter. Beskedet om du har klarat varje projekt eller ej kommer du att få i Blackboard, tillsammans med ordentlig feedback över det som du har gjort bra och det som du inte har gjort tillräckligt bra. I fall du inte lämnar in i tid eller din lösning inte är tillräckligt bra kommer projektet att bli underkänt. För dem som inte har klarat alla projekt kommer det att ges ett tillfälle i slutet på kursen där man ska kunna göra ett försök till.

Duggorna kommer att genomföras i samband med efter-läsningarna: enkla frågor delas ut och svaren samlas in. För de som inte klarar sex av åtta duggor kommer vi att erbjuda möjligheter att examinera omduggorna under examinationsveckorna. Kursen betygsätts enligt det betyg du får på en skriftlig tentamen som erbjuds i slutet av kursen. Man måste anmäla sig på högskolans webbplats för att kunna delta i den skriftliga tentamen.

Sammanfattningsvis examineras kursen genom

1. åtta duggor, av vilka du måste få sex godkända,
2. fyra projekt, av vilka du måste få de tre sista godkända och
3. en skriftlig tentamen, där du måste få betyg 3, 4 eller 5 för att bli godkänd på kursen och betyget på tentan är det betyg du får på kursen.

**Om man blir underkänd** Om man får underkänt på den skriftliga tentamen, finns det ytterligare två tillfällen (omtenta) där tentamen är anpassad till årets kurs. Man kan även komma till tillfällena för nästa års kurs, men då kan innehållet ha förändrats.

## Rekommendationer

Du får 6 timmars undervisning varannan vecka och 5 timmars undervisning varannan vecka men för att klara kursen måste man ofta arbeta mer. Kursen innefattar 7.5 högskolepoäng som betyder ungefär 5 veckors heltidsarbete, d.v.s 200 timmar. Under 16 veckor är detta ungefär 12,5 timmar i veckan (eller 7 timmar självstudier i veckan om man räknar med att du deltar i alla undervisningsmoment vi erbjuder).

Om du undrar något om kursens innehåll eller administration är det bäst att du vänder dig till Verónica Gaspes via epost. Om du undrar något om din utbildning är det bättre att ta kontakt med din studierektor.

## Nyheter för i år

Det är första gången vi erbjuder kursen under en hel termin, därför har vi bland annat infört duggor och efter-läsningar. Vi kommer dessutom att införa två förändringar med anledning av förra årets diskussioner med studenter efter kursvärderingen:

- Vi kommer att arbeta med projekt istället för inlämningar varje vecka av korta programmeringsuppgifter.
- Vi kommer att försöka gruppera studenter med likartade förutsättningar för projektarbete för att underlätta för lärande.

**Lycka till med dina studier!**