



Studieordning for Bacheloruddannelsen i Software

Aalborg Universitet

September 2015

Forord

I medfør Bekendtgørelse af lov 960 af 14. august 2014 om universiteter (Universitetsloven) med senere ændringer fastsættes følgende studieordning for bacheloruddannelsen i Software. Uddannelsen følger endvidere Rammestudieordningen og tilhørende Eksamensordning ved Det Teknisk - Naturvidenskabelige Fakultet.

AAU, september 2014

Lone Leth Thomsen
Studienævnsformand for datalogi

Indholdsfortegnelse

Forord	2
Kapitel 1: Studieordningens hjemmel mv.	3
1.1 Bekendtgørelsesgrundlag.....	3
1.2 Fakultetstilhørsforhold	3
1.3 Studienævnstilhørsforhold.....	3
Kapitel 2: Optagelse, betegnelse, varighed og kompetenceprofil.....	3
2.1 Optagelse.....	3
2.2 Uddannelsens betegnelse på dansk og engelsk.....	3
2.3 Uddannelsens normering angivet i ECTS-point	3
2.4 Eksamensbevisets kompetencebeskrivelse	4
2.5 Uddannelsens kompetenceprofil	4
Kapitel 3: Uddannelsens indhold og tilrettelæggelse	4
1. semester, SW1.....	7
2. semester, SW2.....	13
3. semester, SW3.....	18
4. semester, SW4.....	23
5. semester, SW5.....	27
6. semester, SW6.....	31
Kapitel 4: Ikrafttrædelse, overgangsregler og revision.....	37
Kapitel 5: Andre regler.....	37
5.1 Regler om skriftlige opgaver, herunder bachelorprojektet.....	37
5.2 Regler om merit, herunder mulighed for valg af moduler, der indgår i en anden uddannelse ved et universitet i Danmark eller udlandet.....	38
5.3 Regler omkring forløb og afslutning af bacheloruddannelsen	38
5.4 Afslutning af bacheloruddannelsen.....	38
5.5 Særligt projektforsløb	38
5.6 Eksamensregler	38
5.7 Dispensation	39
5.8 Regler og krav om læsning af tekster på fremmedsprog og angivelse af hvilket kendskab til fremmedsproget(ene) dette forudsætter	39
5.9 Uddybende information	39

Kapitel 1: Studieordningens hjemmel mv.

1.1 Bekendtgørelsesgrundlag

Bacheloruddannelsen i Software er tilrettelagt i henhold til Ministeriet for forskning og uddannelsesbekendtgørelse nr. 1520 af 16. december 2013 om bachelor- og kandidatuddannelser ved universiteterne (Uddannelsesbekendtgørelsen) og bekendtgørelse nr. 670 af 19. juni 2014 om eksamen ved universitetsuddannelser (Eksamensbekendtgørelsen) med senere ændringer. Der henvises yderligere til bekendtgørelse nr. 1487 af 16. december 2013 (Adgangsbekendtgørelsen) og bekendtgørelse nr. 250 af 15. marts 2007 (Karakterbekendtgørelsen) med senere ændringer.

1.2 Fakultetstilhørsforhold

Bacheloruddannelsen hører under Det Teknisk - Naturvidenskabelige Fakultet, Aalborg Universitet.

1.3 Studienævnstilhørsforhold

Bacheloruddannelsen hører under Studienævn for Datalogi ved School of Information and Communication Technology.'

1.4.Censorkorps

Uddannelsen hører under censorkorps for datalogi.

Kapitel 2: Optagelse, betegnelse, varighed og kompetenceprofil

2.1 Optagelse

Optagelse på bacheloruddannelsen i Software forudsætter en gymnasial uddannelse.

Uddannelsens specifikke adgangskrav er Dansk A, Engelsk B og Matematik A, jf. Adgangsbekendtgørelsen.

2.2 Uddannelsens betegnelse på dansk og engelsk

Bacheloruddannelsen giver ret til betegnelsen: Bachelor (BSc) i teknisk videnskab (software). Den engelske betegnelse: Bachelor of Science (BSc) in Engineering (Software)

2.3 Uddannelsens normering angivet i ECTS-point

Bacheloruddannelsen er en 3-årig forskningsbaseret heltidsuddannelse. Uddannelsen er normeret til 180 ECTS.

2.4 Eksamensbevisets kompetencebeskrivelse

Nedenstående vil fremgå af eksamensbeviset:

En bachelor har kompetencer erhvervet gennem et uddannelsesforløb, der er foregået i et forskningsmiljø.

En bachelor har grundlæggende kendskab til og indsigt i sit fags metoder og videnskabelige grundlag. Disse egenskaber kvalificerer bacheloren til videreuddannelse på et relevant kandidatstudium samt til ansættelse på baggrund af uddannelsen.

2.5 Uddannelsens kompetenceprofil

Bacheloruddannelsen i software har som sit mål at give en datalogisk velfunderet indsigt i muligheder for og værktøjer og metoder til konstruktion og analyse af software.

Efter gennemført studium skal en bachelor i software derfor have opnået følgende:

Viden	• har viden om teori, metode og praksis inden for design, konstruktion og analyse af software • kan forstå og reflektere over fagets teorier, metoder og praksis
Færdigheder	• kan anvende den faglige terminologi på korrekt vis • kan anvende metoder og redskaber fra datalogi til at designe, implementere og analysere software • kan vurdere teoretiske og praktiske problemstillinger inden for software og begrunde og vælge relevante løsningsmodeller ud fra kendskab til de muligheder og begrænsninger, som er givet af datalogiens teorier og metoder
Kompetencer	• kan formidle problemstillinger og løsningsmodeller til såvel fagfæller som ikke-specialister, samarbejdspartnere og brugere. • kan håndtere komplekse og udviklingsorienterede situationer i studie- og arbejdssammenhænge. • kan selvstændigt indgå i fagligt og tværfagligt samarbejde med en professionel tilgang. • kan identificere egne læringsbehov og strukturere egen læring i forskellige læringsmiljøer.

Der henvises i øvrigt til rammestudieordningens almene akademiske kompetencemål for bacheloruddannelser (3.2.1).

Kapitel 3: Uddannelsens indhold og tilrettelæggelse

Bacheloruddannelsen i software er tilrettelagt som et problembaseret studium, baseret på projektmoduler og kursusmoduler. I projektmodulerne er undervisningsformen projektarbejde; i kursusmodulerne kan der anvendes en blanding af forelæsninger, opgaveløsning, workshops og andre undervisningsformer, der fastlægges af underviserne.

Uddannelsesoversigt:

Alle moduler bedømmes gennem individuel gradueret karakter efter 7-trinsskalaen *eller* bestået/ikke bestået. Alle moduler bedømmes ved ekstern prøve (ekstern censur) eller intern prøve (intern censur eller ingen censur).

Semester	Modul	ECTS	Bedømmelse	Prøve
1.	Hvis programmer er løsningen – hvad er så problemet?	5	Bestået/ikke bestået	Intern
	Fra eksisterende software til modeller	10	7-trinsskala	Intern
	Lineær algebra	5	7-trinsskala	Intern
	Problembaseret læring i videnskab, teknologi og samfund	5	Bestået/ikke bestået	Intern
	Imperativ Programmering	5	7-trinsskala	Intern
2.	Programmering og Problemløsning	15	7-trinsskala	Ekstern
	Diskret matematik	5	7-trinsskala	Intern
	Computer arkitektur	5	7-trinsskala	Intern
	Objektorienteret programmering	5	7-trins-skala	Intern
3.	Udvikling af applikationer – fra brugere til data, algoritmer og test – og tilbage igen	15	7-trins-skala	Intern
	Systemudvikling	5	7-trins-skala	Intern
	Design og evaluering af brugergrænseflader	5	7-trins-skala	Intern
	Algoritmik og datastrukturer	5	7-trins-skala	Ekstern
4.	Design, definition og implementation af programmeringssprog	15	7-trins-skala	Ekstern
	Syntaks og semantik	5	7-trins-skala	Ekstern
	Principper for styresystemer og parallelitet	5	7-trins-skala	Ekstern
	Sprog og oversættere	5	7-trins-skala	Ekstern
5.	Indlejrede systemer	15	7-trins-skala	Intern
	Maskinintelligens (valgfri)	5	7-trins-skala	Intern
	Tidstro software (valgfri)	5	7-trins-skala	Intern
	Software Engineering	5	7-trins-skala	Ekstern
	Beregnelighed og kompleksitet	5	7-trins-skala	Ekstern
6.	Bachelor projekt	15	7-trins-skala	Ekstern
	Avancerede algoritmer (valgfri)	5	7-trins-skala	Intern
	Semantik og verifikation (valgfri)	5	7-trins-skala	Intern

	Databasesystemer	5	7-trins-skala	Ekstern
	Videnskabsteori	5	Bestået/ikke bestået	Intern
Sum		180		

1. semester, SW1

Titel:	Hvis programmer er løsningen - hvad er så problemet? (P0) (If Programs are the Solution – then what is the Problem?)
Omfang:	5 ECTS (Projektmodul)
Forudsætninger:	Optagelse på 1. semester.
Formål:	At den studerende opnår viden om problemorienteret projektarbejde og specifikt kan analysere og definere et problem inden for programmer samt beskrive problemstillingen i en anvendelsessammenhæng med vægt på enten en teknisk eller social sammenhæng.
Begrundelse:	Projektarbejdet fokuserer på at opnå tidlig erfaring med problemorienteret projektarbejde i grupper for at opbygge kompetence til P1-projektet.
Mål:	<u>Viden:</u> Den studerende skal kunne forstå og gøre rede for de i projektet anvendte teorier og metoder. <u>Færdigheder:</u> Den studerende skal kunne: • formidle analyse og afgrænsning af en problemstilling inden for software • formidle projektets overvejelser, arbejdsresultater og arbejdsprocesser skriftligt, grafisk og mundtligt • beskrive opnåede erfaringer med gruppens projektarbejde <u>Kompetencer:</u> Den studerende skal kunne: • afgrænse en problemstilling inden for software • beskrive og analysere en problemstilling samt vælge og forsvare en problemformulering • foreslå og argumentere for mulige løsninger af et formuleret problem.
Indhold:	Som dokumentation for projektarbejdet skal projektgruppen: • udarbejde en P0-rapport, og • udarbejde en P0-procesanalyse. Efter aflevering af projektrapporten afholdes en erfaringsopsamling, hvor et antal P0- projektgrupper fremlægger deres erfaringer med projektgruppens arbejdsproces. Erfaringsopsamlingen danner grundlag for den enkelte gruppes procesanalyse.
Prøveform:	Intern mundtlig prøve på baggrund af projektdokumentationen
Bedømmelse:	Individuel bedømmelse, bestået/ikke-bestået
Vurderingskriterier:	Se Rammestudieordningen

Titel:	Fra eksisterende software til modeller (P1) (From Existing Software to Models)
Omfang:	10 ECTS (Projektmodul)
Forudsætninger:	Projektmodulet P0 samt at kurserne Imperativ programmering og POPBL følges parallelt med projektarbejdet.
Formål:	At den studerende opnår færdigheder i problemorienteret projektarbejde i en gruppe samt viden om sammenhænge mellem problemdefinition, modeldannelsers rolle i forståelse og konstruktion af programmer, og programmer som løsning på et problem i en problemstillings kontekst. Endvidere at opnå viden om fagets indhold og fagets videre potentialer.
Begrundelse:	Projektarbejdet fokuserer på at opnå erfaring med problemorienteret projektarbejde i tilknytning til programmering og programforståelse for at opbygge både software- og projektkompetence til P2 projektet.
Mål:	<u>Viden:</u> Den studerende skal kunne: • forstå og gøre rede for de i projektet anvendte teorier og metoder til analyse af den valgte problemstilling, • specielt forstå og gøre rede for de begreber inden for programmering og modellering, som er blevet anvendt i forbindelse med projektet, • forstå og gøre rede for projektets kontekstuelle forhold. <u>Færdigheder:</u> Den studerende skal kunne: • vælge, beskrive og anvende en af de i HOPL-kurset foreslåede metoder til organisering af gruppensamarbejdet og til løsning af eventuelle gruppekonflikter, • anvende begreber og værktøjer til problembaseret projektarbejde og reflektere skriftligt over den problembaserede læring i projektsammenhæng • formidle projektets arbejdsresultater og arbejdsprocesser på en struktureret og forståelig måde, såvel skriftligt, grafisk som mundtligt. <u>Kompetencer:</u> Den studerende skal kunne: • analysere en problemstilling inden for software og inden for denne problemstilling formulere et problem, hvor programmering kan indgå som del af løsningen • opstille en model af problemstillingen • inddrage relevante begreber og metoder til analyse og vurdering af projektets løsninger i relation til problemets kontekst
Indhold:	Som del af projektet skal gruppen i fællesskab udarbejde et mindre program af høj kvalitet. Der skal i denne forbindelse også være en beskrivelse af væsentlige egenskaber ved programmet.

Som dokumentation for projektarbejdet skal projektgruppen:

- udarbejde en P1-projektrapport,
- udarbejde et nyt P1-projektforslag, som vil kunne præsenteres ved næste P1-forløb
- deltage i P1-erfaringsopsamling,
- udarbejde en P1-procesanalyse

Midt i projektperioden afholdes et statusseminar, hvor projektgruppen fremlægger sin problemformulering, arbejdsresultater og erfaringer med projektarbejdsprocessen. Ved dette seminar deltager mindst én anden projektgruppe og de pågældende gruppers vejledere.

Efter aflevering af projektrapporten afholdes en erfaringsopsamling, hvor et antal P1-projektgrupper fremlægger deres erfaringer med projektets arbejdsproces. Erfaringsopsamlingen danner grundlag for den enkelte gruppes procesanalyse.

Prøveform: Intern mundtlig prøve på baggrund af projektdokumentationen

Bedømmelse: Individuel karakter efter 7-trins-skalaen

Vurderingskriterier: Se Rammestudieordningen

Titel: **Lineær algebra
(Linear Algebra)**

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger:

Mål: Viden:

Den studerende skal have:

- viden om definitioner, resultater og teknikker indenfor teorien for lineære ligningssystemer
- kendskab til lineære transformationer og deres sammenhæng med matricer
- viden om computerværktøjet Matlab og dets anvendelse indenfor lineær algebra
- kendskab til simple matrixoperationer
- kendskab til invertibel matrix og invertibel lineær afbildning
- kendskab til vektorrummet R^n og underrum deraf
- kendskab til lineær afhængighed og uafhængighed af vektorer, samt dimension og basis for underrum
- kendskab til determinant for matricer
- kendskab til egenværdier og egenvektorer for matricer og deres anvendelse
- kendskab til projektioner og ortonormale baser

- viden om første ordens differentiaalligninger, samt om systemer af lineære differentiaalligninger

Færdigheder:

Den studerende skal kunne:

- anvende teori og regneteknik for lineære ligningssystemer til at afgøre løsbarehed, og til at bestemme fuldstændige løsninger og deres struktur
- repræsentere lineære ligningssystemer ved hjælp af matrixligninger, og omvendt
- bestemme og anvende reduceret echelonform af en matrix
- anvende elementære matricer i forbindelse med Gauss-elimination og inversion af matricer
- afgøre lineær afhængighed eller lineær uafhængighed af små systemer af vektorer
- bestemme dimension af og basis for underrum
- bestemme matrix for en givet lineær afbildning, og omvendt
- løse simple matrixligninger
- beregne invers af små matricer
- bestemme dimension af og basis for nulrum og søjlerum
- beregne determinanter og kunne anvende resultatet af beregningen
- beregne egenværdier og egenvektorer for simple matricer
- kunne afgøre, om en matrix er diagonaliserbar, og i bekræftende fald gennemføre en diagonalisering, for simple matricer
- beregne den ortogonale projektion på et underrum af $\mathbb{R}^n$
- løse separable og lineære første ordens differentiaalligninger, generelt, og med begyndelsesbetingelser

Kompetencer:

Den studerende skal:

- udvikle og styrke sit kendskab til, forståelse af, og anvendelse af matematiske teorier og metoder indenfor andre fagområder
- ud fra givne forudsætninger kunne ræsonnere og argumentere med matematiske begreber indenfor lineær algebra

Undervisningsform: Forelæsninger med tilhørende opgaveregning.

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Individuel karakter efter 7-trins-skalaen

Vurderingskriterier: Er angivet i Rammestudieordningen

**Titel: Problembaseret læring i videnskab, teknologi og samfund
(Problem-based Learning in Science, Technology and Society)**

Omfang: 5 ECTS (Kursusmodul)

Formål: Kursets formål er at støtte de studerende inden for datalogi og software, teoretisk såvel som praktisk i at planlægge og udføre et problembaseret projektarbejde i grupper, under hensyntagen til de tilgængelige ressourcer og den samfundsmæssige sammenhæng. Brug og udvikling af informationsteknologier og software er påvirket af og påvirker menneskers og samfunds udvikling og er genstand for kurset.

Mål: Efter kurset skal den studerende have:

Viden, der gør den studerende i stand til at:

- redegøre for grundlæggende læringsteori;
- redegøre for teknikker til planlægning og styring af projektarbejde;
- redegøre for forskellige tilgange til problembaseret læring; herunder Aalborg-modellens udgangspunkt i problemer, der indgår i en samfundsmæssig og/eller humanistisk sammenhæng.
- redegøre for forskellige tilgange til analyse og vurdering af problemstillinger inden for udvikling og brug af software i et videnskabsteoretisk, etisk, og samfundsmæssigt perspektiv;
- redegøre for konkrete metoder til at udføre analyse og vurdering; herunder vurdering af brugbarhed af software og de sammenhænge i hvilke software bliver udviklet herunder brugerinddragelse

Færdigheder, der gør de studerende i stand til at:

- planlægge og styre et problembaseret studieprojekt;
- analysere projektgruppens organisering af gruppesamarbejdet, med henblik på at identificere stærke og svage sider og forslå forbedringer;
- reflektere over årsager til og anvise mulige løsninger på eventuelle gruppekonflikter;
- analysere og vurdere egen studieindsats og læring, med henblik på at identificere stærke og svage sider, og der ud fra overveje videre studieforløb og studieindsats;
- reflektere over de anvendte metoder i et videnskabsteoretisk perspektiv
- udpege relevante fokusområder, begreber og metoder til at vurdere og udvikle løsninger under hensynstagen til de kontekstuelle sammenhænge i hvilke løsningen skal indgå
- reflektere over sit professionelle virke i relation til det omgivende samfund.

Kompetencer, som gør den studerende i stand til at:

- indgå i en projektorganiseret problemløsning;
- formidle resultaterne af projektarbejde;
- reflektere og udvikle egen læring;
- indgå i projektorganiserede læreprocesser;

Indhold: Kursets indhold sigter ligeligt på projektgruppens arbejdsform og den kontekstuelle helhed for projektet.

Kurset kan omfatte:

- Studieintroduktion og -teknik;
- Videnskabelig redelighed;
- Skriftlig og mundtlig formidling af projekresultater.
- Erfaringsopsamling
- Introduktion til planlægning, styring og ledelse af læringsprojekter
- Introduktion til styring af softwareudviklingsprojekter herunder introduktion til teknikker som Scrum
- Kommunikationen i og udad gruppen
- Læringsstile, teamroller og gruppedynamik;
- Kreativitet i projektarbejdet
- Konflikthåndtering;
- Introduktion til teori om læreprocesser;
- Introduktion til videnskabsteori - herunder datalogiens videnskabsteori
- Introduktion til sociologisk metode, kvalitativ og kvantitativ undersøgelse;
- Faser i et problemorienteret projektarbejde fra initierende problem over problemanalyse til problemformulering;
- Helhedsvurdering af videnskaben/teknologier/produktet i relation til brugerne og samfund, herunder:
 - Miljø, forbrug og socialt ansvar;
 - Samfundsøkonomi; herunder softwaresystemers samfundsøkonomiske betydning
 - Introduktion til kulturforståelse og interkulturel kommunikation
 - Brugbarhed og nytte af softwaresystemer;
 - Introduktion til politiske processer, magt og regulering; herunder inddragelse af interessegrupper i forbindelse med softwareudvikling
- Metoder til analyse og dokumentation af gruppens læreprocesser;

Undervisningsform: Kurset er organiseret som en blanding af forelæsninger, seminarer, workshops, gruppekonsultation og selvstudie

Prøveform: Intern individuel skriftlig prøve

Kurset eksamineres individuelt på baggrund af en skriftlig opgave (max. 5 sider), som tillæg til den skriftlige procesanalyse og projektafgrænsningen indeholdende en påpegning men samtidig en afgrænsning fra at analysere relevante kontekstuelle sammenhænge. Den skriftlige opgave skal indeholde en personlig refleksion over projektets proces og en analyse af den individuelle læreproces (max. 3 sider) samt en overordnet vurdering af projektets produkt i relation til de påpegede kontekstuelle sammenhænge (max. 2 sider).

Bedømmelse: Individuel bedømmelse, bestået/ikke-bestået

Vurderingskriterier: Se Rammestudieordningen

Titel:	Imperativ Programmering (Imperative Programming)
Omfang:	5 ECTS (Kursusmodul)
Forudsætninger:	Ingen
Formål:	I dette kursus opnår den studerende indblik i grundlæggende begreber som algoritmer, datastrukturer og computerarkitekturer.
Begrundelse:	Computere er – uanset fagområde – et af de vigtigste værktøjer til problemløsning i dag. Den studerende skal derfor opnå et kendskab til datalogiske grundbegreber i så almen en form, at vedkommende bliver i stand til at løse problemer ved hjælp af imperative programmeringssprog.
Mål:	<u>Viden:</u> Den studerende skal forstå grundbegreberne inden for følgende teorier og metoder: • Udviklingsmiljø og kompilering • Imperative principper • Datatyper og variable • Kontrolstrukturer • Funktioner og procedurer • Datastrukturer herunder arrays • Input/output • Sammensatte datastrukturer • Simple algoritmer (f.eks. sortering og søgning) • Basal test af programmer <u>Færdigheder:</u> Den studerende skal efter kurset være i stand til at: • skrive, afvikle og teste programmer hvori de ovennævnte grundbegreber indgår i løsningen • anvende korrekt fagterminologi <u>Kompetencer:</u> • Den studerende kan efter kurset både selvstændigt og i samarbejde med andre implementere et imperativt program som løsning på en defineret opgave.
Prøveform:	Intern mundtlig individuel prøve
Bedømmelse:	Individuel bedømmelse, efter 7-trins-skalaen
Vurderingskriterier:	Er angivet i rammestudieordningen

2. semester, SW2

Titel:	Programmering og problemløsning (P2) (Programming and Problem Solving)
Omfang:	15 ECTS (Projektmodul)
Forudsætninger:	Projektmodulerne på 1.semester, kursusmodulerne Imperativ programmering og POPBL samt at kursusmodulerne Diskret matematik, Computer arkitektur og Objektorienteret programmering følges parallelt med projektarbejdet.
Formål:	At den studerende lærer hvordan man i problemløsning som en væsentlig del kan benytte sig af programmering og dertil hørende modeldannelse og opnår yderligere erfaring i problemorienteret projektarbejde i en gruppe.
Begrundelse:	Projektarbejdet fokuserer på at opnå færdigheder i programmering og dertil hørende modeldannelse i forbindelse med problemløsning. Projektarbejdet skal således sikre, at de studerende opnår et fælles fundament i programmering, der kan udnyttes i efterfølgende semestre. Projektarbejdet fokuserer desuden på at opnå erfaring med problemorienteret projektarbejde i tilknytning til programmering.
Mål:	<u>Viden:</u> Den studerende skal efter gennemført projektmodul kunne: • forstå og gøre rede for syntaks og adfærd af sproglige konstruktioner i et konkret programmeringssprog og hvordan de kan anvendes <u>Færdigheder:</u> Den studerende skal efter gennemført projektmodul kunne: • implementere et større program og herunder udnytte og forstå begreber, strukturer og faciliteter i programmeringssproget samt på denne baggrund konstruere et velstruktureret program • gennemføre en afestning af programmet som - i en rimelig grad - sikrer, at det udviklede program svarer til problemformuleringen. <u>Kompetence:</u> Efter gennemførelsen af projektmodulet skal den studerende kunne: • afgrænse en problemstilling og inden for denne problemstilling formulere og forsvare et problem, der kan løses ved brug af programmering som en væsentlig del af løsningsmetoden • udarbejde en model, der kan anvendes i forbindelse med løsningen af det valgte problem • fuldføre arbejdet frem til et køredygtigt og demonstrerbart program og herunder vise evne til afgrænsning • beskrive hvordan programmet løser et formuleret problem inden for problemstillingens kontekst • inddrage relevante brugsmæssige og sociale forhold af løsninger inden for datalogi og softwareteknologi • beskrive, reflektere over og analysere de opnåede erfaringer med problemorienteret projektarbejde i en gruppe

Indhold: Som del af projektets problemløsning skal gruppen i fællesskab udarbejde et større program af høj kvalitet. Der skal i denne forbindelse også være en beskrivelse af væsentlige egenskaber ved programmet. Der skal specielt gøres rede for det overordnede design af løsningen, eventuelt med hovedvægt på programstrukturer og anvendte algoritmer.

Som dokumentation for projektarbejdet skal projektgruppen

- udarbejde en P2-rapport,
- udarbejde en P2-procesanalyse,

Midt i projektperioden afholdes et statusseminar, hvor projektgruppen fremlægger sin problemformulering, arbejdsresultater og erfaringer med projektarbejdsprocessen. Ved dette seminar deltager mindst én anden projektgruppe og de pågældende gruppers vejledere.

Prøveform: Ekstern mundtlig prøve på baggrund af projektrapport

Bedømmelse: Individuel karakter efter 7-trins-skalaen

Vurderingskriterier: Se Rammestudieordningen

Titel: **Diskret matematik
(Discrete Mathematics)**

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger: Kursusmodulet Lineær algebra (1. semester)

Mål: Viden:
Den studerende skal opnå viden om følgende teorier og metoder:

- Mængdelære: Mængder, relationer, funktioner, partielle ordninger, ækvivalensrelationer
- Grundlæggende talteori: Modulær aritmetik. Euklids algoritme. Den kinesiske restsætning. Fermats lille sætning. Primtalsopløsning.
- De rationale tals tællelighed.
- Rekursive/iterative algoritmer. Tidskompleksitet.
- Asymptotisk notation. Logaritme og eksponentialfunktioner med grundtal 2. Store-O-notationen.
- Kombinatorik: Binomialformlen.
- Rekursive funktioner. Rekurrensligninger.
- Bevisteknikker: Svag og stærk induktion. Modstridsbevis, bevis ved kontraposition, konstruktivt bevis.
- Logisk notation: Udsagnslogik, kvantorer.
- Grafteori: Orienterede og ikke-orienterede grafer. Veje, stier, træer. Grafalgoritmer. Søgning i grafer. Korteste vej.

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- Kunne gennemføre beviser for resultater indenfor kursets emner ved hjælp af de i kurset behandlede bevisteknikker.
- Kunne gøre brug af de fornødne skriftlige færdigheder i disse sammenhænge.

Kompetencer:

Den studerende skal kunne anvende begreber og teknikker fra diskret matematik, herunder i sammenhænge, hvor algoritmer indgår.

Undervisningsform: 20 forelæsninger med tilhørende opgaveregning. Desuden 5 kursusgange, hvor der arbejdes med større skriftlige opgaver.

Prøveform: 4 timers skriftlig eksamen uden brug af computeralgebra-værktøj; med intern censur

Bedømmelse: Individuel karakter efter 7-trins-skalaen

Vurderingskriterier: Se Rammestudieordningen

Titel: **Computerarkitektur
(Computer Architecture)**

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger: Kursusmodulet Imperativ Programmering

Mål: I dette kursus opnår den studerende kendskab til grundlæggende begreber som kørende programmer, instruktionssæt, mikrokoder, køretids miljøer, digitale kredsløb, mm.

Begrundelse: Grundlæggende set er computerprogrammer skrevet i højniveausprog, oversat til maskinkode og dernæst afviklet på datamater. Den studerende skal derfor som minimum opnå et kendskab til de forskellige trin i oversættelser og afvikling af højniveaucomputerprogrammer på datamater, heriblandt hvordan en konkret processor afvikler et computerprogram.

Mål: Viden:
Den studerende skal opnå viden om følgende teorier og metoder:

- Digitale kredsløb.
- Den fysiske processors opbygning.
- Harvard- og von Neumann-arkitektur.
- Mikrokoder.
- ISA-niveau.
- Hukommelse deriblandt cache-memory.
- Heltal og flydende talberegninger og repræsentation.
- Køretidsmiljøet for et kørende program.
- Operativsystemer og proceshåndtering.

Færdigheder:

Den studerende skal efter kurset være i stand til at:

- Kunne implementere et simpelt program ved hjælp af instruktioner på assemblerniveau og uden brug af højniveauoversættelse.
- Kunne forstå og udnytte specielle instruktioner som f. eks stak-ændringsinstruktioner til at gemme og håndtere midlertidige data.
- Anvende korrekt fagterminologi.

Kompetencer:

Den studerende kan efter kurset både selvstændigt og i samarbejde med andre formulere og løse simple lavniveau-relaterede problemstillinger som er funderet i viden om kredsløb, køretidsmiljøer, operativsystemer, mm.

Prøveform: Intern individuel mundtlig eller skriftlig prøve

Bedømmelse: Individuel bedømmelse efter 7-skala

Vurderingskriterier: Se Rammestudieordningen

Titel: **Objektorienteret programmering
(Object-oriented Programming)**

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger: Kursusmodulet Imperativ programmering.

Formål: At den studerende lærer de væsentlige begreber og struktureringsmekanismer inden for objektorienterede programmeringssprog og opnår færdigheder inden for programmering i et sprog inden for dette paradigme.

Begrundelse: Objektorienteret programmering er et dominerende programmeringsparadigme i softwareudvikling.

Mål: Viden:
Den studerende skal opnå forståelse af teorier og metoder inden for det objektorienterede programmeringsparadigme, og her specielt følgende aspekter:

- begreber og begrebsdannelse inden for objektorientering
- klasser og objekter
- datatilgang, properties og metoder
- førsteklasses-metoder
- collection-klasser
- specialisering, udvidelse og nedarvning
- polymorfi og dynamisk binding
- nedarvning
- interfaces og abstrakte klasser
- exception handling
- generiske typer og metoder
- kontrakter og assertions

Færdigheder:

Den studerende skal kunne:

- programmere i et objektorienteret programmeringssprog, således at disse sprogs centrale egenskaber bliver anvendt
- forklare og argumentere for sammenhænge og detaljer i et objektorienteret program
- udarbejde og gennemføre en systematisk afestning af et objektorienteret program

Kompetencer:

Den studerende skal kunne udforme og dokumentere et objektorienteret program, således at det kan køres og er forståeligt, læsbart, og tilgængeligt for andre programmører.

Undervisningsform: En blanding af forelæsninger, mindre øvelser og en større opgave. I den større opgave skal de studerende, alene eller i mindre grupper, udvikle et objektorienteret program, som skal løse et på forhånd formuleret og afgrænset problem. Kursusholderen fastsætter den maksimale gruppestørrelse og definerer et konkret område, inden for hvilket problemet for den større opgave skal findes. De studerende har en vis frihed til at påvirke problemformuleringen. Programmet og dets dokumentation skal afleveres i udskrift og på elektronisk form.

Prøveform: Intern mundtlig prøve med udgangspunkt i besvarelsen af den større opgave

Bedømmelse: Intern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

3. semester, SW3

Titel: Udvikling af applikationer – fra brugere til data, algoritmer og test – og tilbage igen
(Developing Applications – from Users to Data, Algorithms and Tests – and Back Again)

Omfang: 15 ECTS (Projektmodul)

Forudsætninger: Projekt- og kursusmodulerne på 1. – 2. semester (SW1-2)

Formål: Den studerende skal opnå viden om problemstillinger og fundamentale teknikker i udvikling af applikationer til løsning af realistiske opgaver; og opnå erfaring med udvikling af store systemer, arbejdsdeling og kvalitetskontrol herunder afestning og afprøvning

Begrundelse: Projektenheden fokuserer på at opnå færdigheder med bestemte og på forhånd fastlagte metoder, sprog og værktøjer. Projektenheden skal således sikre, at den studerende opnår et fælles fundament omkring programudvikling, der kan udnyttes i efterfølgende semestre

Mål:	<u>Viden:</u> Den studerende skal efter gennemført projektmodul kunne: • analysere og modellere krav i det objektorienterede paradigme • strukturere en applikation i en flerlags arkitektur og ved hjælp af gængse program mønstre • designe, realisere og af teste en applikation i det objektorienterede paradigme • forstå og udnytte begreber og faciliteter i paradigmet og på den baggrund konstruere en applikation af høj, intern og ekstern kvalitet <u>Færdigheder:</u> Den studerende skal efter gennemført projektmodul kunne: • gennemføre systematisk af testning af applikationen og påvise at applikationen svarer til intentioner og brugernes behov • gennemføre systematisk evaluering af brugergrænsefladen • argumentere for trufne valg i alle udviklingsprocessens aktiviteter, herunder forklare krav, arkitektur og hvordan brugeres behov hænger sammen <u>Kompetence:</u> Efter gennemførelsen af projektmodulet skal den studerende kunne: • udvikle en kørende applikation som løser brugernes problem • beskrive og reflektere over den anvendte arbejdsform i udviklingsprojektet
Undervisningsform:	Projektarbejde, der skal omfatte: • formulering, analyse og bidrag til løsning af et aktuelt problem, normalt inden for det område af datalogi, som er emnet for projektmodulet på 3. semester
Prøveform:	Mundtlig prøve på baggrund af projektrapport
Bedømmelse:	Intern bedømmelse efter 7-trins-skala
Vurderingskriterier:	Se Rammestudieordningen

Titel:	Systemudvikling (Systems Development)
Omfang:	5 ECTS (Kursusmodul)
Forudsætninger:	Kursusmodulet Objektorienteret programmering (2. semester) samt projektmodulerne på SW1 og SW2
Mål:	<u>Viden:</u> Den studerende skal opnå viden om følgende teorier og metoder: Objektorienteret modellering i analyse og design: • modellering af kontekst (anvendelsesområde og problemområde)

- objektorienterede begreber: klasse, objekt, hændelse, struktureringsformer, funktion, brugsmønstre, komponenter, komponentarkitektur
- UML: klassediagram, tilstandsændringsdiagram, sekvensdiagram, brugsmønstrediagram

Modellering med mønstre:

- mønstre til modellering af anvendelsesområder og problemområder
- mønstre til sammensætning af komponenter
- specielt analysemønstrene: genstand-beskrivelse, hierarki, trinvis-rolle, materiale, procedure
- specielt designmønstrene: samling, lagdelt, observatør, klient-server, model-view-controller

Systemudviklingsmetode:

- vandfaldsmetode og model-drevet udvikling
- iterative metode og prototype-drevet udvikling
- aktiviteter i systemudvikling og sammenhænge mellem aktiviteter

Systemudviklingspraksis:

- teknikker til fastlæggelse af konkret metode
- relationen mellem metode og praksis
- styrker og svagheder ved model-drevet og ved prototype-drevet udvikling

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- kunne redegøre præcist og ved brug af fagets begreber og modelleringssprog
- kunne modellere krav til et system, dets kontekst og alle dets forskellige dele (model, funktioner og græseflader)
- kunne modellere et systemdesign på komponentniveau samt beskrive sammenhæng mellem komponenter

Kompetencer:

Den studerende skal kunne anvende begreberne, mønstrene og modelleringssproget til at beskrive et konkret system som løser en veldefineret opgave

Undervisningsform: Kursus

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Intern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

**Titel: Design og evaluering af brugergrænseflader
(Design and Evaluation of User Interfaces)**

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger: Kursusmodulet Objektorienteret programmering (2. semester) samt projektmodulerne på SW1 og SW2

Mål: Viden:
Den studerende skal opnå viden om følgende teorier og metoder:

Fundamentale menneske-maskin interaktion:

- interaktionsdesign
- usability og user experience
- designprincipper
- interaktionsformer
- menneskelig kognition, perception og hukommelse

Interaktionsdesignprocessen:

- aktiviteter i interaktionsdesign
- user-centred design
- contextual design og participatory design
- forskellige livscyklusmodeller til interaktionsdesign

Brugskontekst og brugere:

- forståelse af behov og krav: f.eks. interview, observation, spørgeskema, probes, kortsortering
- opgaveanalyse: f.eks. hierarkisk opgaveanalyse, mål, opgaver, handlinger
- scenarier og personas
- brugsmønstre

Design af grænseflader:

- visuelle designsprincipper
- gestaltlove
- skitsering og prototyping
- konceptuel og fysisk brugergrænsefladedesign

Usabilityevaluering:

- aktiviteter
- roller og opgaver
- identifikation af usabilityproblemer

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- kunne forstå basale og avancerede begreber og teorier om menneske-maskin interaktion
- kunne redegøre præcist for og forklare aktiviteterne i designet af en brugergrænseflade
- kunne forklare og redegøre for aktiviteterne i en usabilityevaluering

Kompetencer:

Den studerende skal kunne anvende begreber, teknikker og metoder til at designe og evaluere et konkret system som løser en veldefineret opgave

Undervisningsform: Kursus

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Intern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

**Titel: Algoritmik og datastrukturer
(Algorithmics and Data Structures)**

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger: Første år af bacheloruddannelsen i software.

Mål: Viden:

Den studerende skal opnå viden om følgende teorier og metoder:

- matematiske grundbegreber såsom rekursion, induktion, konkret og abstrakt kompleksitet
- interne og eksterne datastrukturer, algoritmeprincipper såsom søgning, søgetræer, intern og ekstern sortering, dynamisk programmering, del-og-indtag
- grafer og grafalgoritmer såsom korteste vej, sammenhængskomponenter, udspændende træer

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- bestemme abstrakte kompleksitet for konkrete funktioner
- gennemføre kompleksitets- og korrekthedsanalyse på simple algoritmer, herunder rekursive algoritmer
- udvikle og anvende passende algoritmer til standard-opgaver, som f.eks. søgning, sortering og vejfinding

Kompetencer:

Den studerende skal, stillet overfor en ikke-standard programmeringsopgave kunne

- udvikle algoritmer og datastrukturer til løsning af opgaven
- analysere de udviklede algoritmer

Undervisningsform: Kursus

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Ekstern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

4. semester, SW4

Titel: Design, definition og implementering af programmeringssprog
(Design, Definition and Implementation of Programming Languages)

Omfang: 15 ECTS (Projektmodul)

Forudsætninger: Projekt- og kursusmodulerne på 1. – 3. semester samt at kursusmodulerne Syntaks og semantik, Sprog og oversættere og Principper for samtidighed og styresystemer følges parallelt med projektarbejdet

Formål: At den studerende lærer, hvordan man kan designe og implementere et programmeringssprog og hvordan denne proces kan understøttes af formelle definitioner af sprogets syntaks og semantik og teknikker og metoder til oversætterkonstruktion

Begrundelse: Al software er skrevet i et programmeringssprog og oversættes eller fortolkes for at kunne eksekveres. Design, beskrivelse og konstruktion af Programmeringssprog, oversættere, fortolkere og lignende værktøjer er af den grund centrale emner i datalogi. Projektarbejdet fokuserer derfor på at sikre, at de studerende forstår vigtige underliggende begreber i programmeringssprogenes verden, hvorfor disse begreber er opstået og hvordan de beskrives formelt og repræsenteres i en implementation. Forståelse af disse emner er fundamentale i forståelsen af nye og eksisterende programmeringssprog og deres anvendelsesmuligheder. Ydermere anvendes mange teknikker og værktøjer, oprindeligt udviklet til sprog og oversættere, også i andre sammenhænge i programudvikling.

Mål: Viden:
Den studerende skal efter gennemført projektmodul kunne:

- forstå og gøre rede for grundbegreberne i en formel definition af et programmeringssprogs syntaks og semantik
- dokumentere kendskab til og overblik over de berørte teknikker og begreber inden for sprogdesign og oversætterkonstruktion
- redegøre for de enkelte faser og sammenhængen mellem faserne i en oversætter
- redegøre for de anvendte implementationsteknikker i den konstruerede oversætter/fortolker
- benytte korrekt fagterminologi

Færdigheder:

Den studerende skal efter gennemført projektmodul kunne:

- beskrive et programmeringssprogs syntaks og semantik ved brug af relevante metoder til formelle definition
- implementere en oversætter eller fortolker til et konkret programmeringssprog eller til en udvidelse til et eksisterende programmeringssprog
- ræsonnere datalogisk om og med de berørte begreber og teknikker

Kompetence:

Den studerende skal efter gennemført projektmodul kunne:

- vurdere anvendelse og anvendelighed af kendte værktøjer og teknikker til definition og implementation af programmeringssprog
- forstå og gøre rede for hvordan konkrete sproglige begreber repræsenteres på køretidspunkter og i formel semantik

Undervisningsform: Projektarbejde, der skal omfatte:

- en analyse af en datalogisk problemstilling, hvis løsning naturligt kan beskrives i form af design af et konkret programmeringssprog
- en formel definition af relevante, centrale dele af dette programmeringssprogs syntaks og semantik
- i tilknytning hertil konstruktion af en oversætter/fortolker for sproget

Prøveform: Mundtlig prøve på baggrund af projektrapport

Bedømmelse: Ekstern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

Titel: Syntaks og semantik
(Syntax and Semantics)

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger: Kursusmodulerne Diskret Matematik (2. semester) og Algoritmik og datastrukturer (3. semester)

Mål: Viden:
Den studerende skal opnå viden om følgende teorier og metoder:

Formel sprogteori:

- Teorier for beskrivelse og genkendelse af regulære sprog: Deterministiske og nondeterministiske endelige automater, regulære udtryk og ækvivalens af disse
- Teorier for beskrivelse og genkendelse af kontekstfrie sprog: Kontekstfrie grammatikker og pushdown-automater og ækvivalens af disse.
- Begrænsninger ved regulære og kontekstfrie sprog: Pumping Lemma for regulære og kontekstfrie sprog

Semantik af programmeringssprog:

- Strukturel operational semantik: Big-step og small-step semantik af gængse programmeringskonstruktioner. Semantisk ækvivalens. Semantik af scope-regler og parametermekanismer. Semantik af parallelitet og nondeterminisme
- Rekursive definitioner og beregning af fikspunkter

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- Kunne redegøre præcist og ved brug af fagets terminologi og notation for resultater inden for formel sprogteori og semantik af programmeringssprog og hvordan og i hvilket omfang disse resultater kan anvendes
- Kunne gøre brug af de fornødne skriftlige færdigheder i disse sammenhænge

Kompetencer:

Den studerende skal kunne anvende begreber og teknikker fra formel sprogteori og semantik af programmeringssprog, herunder i design og beskrivelse af programmeringssprog

Undervisningsform: Kursus

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Ekstern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

Titel: **Principper for styresystemer og parallelitet**
(Principles of Operation Systems and Concurrency)

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger: Kursusmodulet Computer arkitektur (2. semester)

Mål: Viden:

Den studerende skal opnå viden om følgende teorier og metoder:

- processer og tråde: anvendelse, realisering, tilstandsmodeller, multi-threading, programmering med tråde, proces-/tråd-oprettelse og tidsplanlægning
- filsystemer: navnerum, realisering af filsystemer, strategier for pladsfordeling
- hukommelseshåndtering, tildeling i primært lager: fast inddeling, virtuel hukommelse, sidedelt hukommelse, sideerstatningsalgoritmer, delt lager, copy-on-write, demand paging, rammetildeling

- styresystemkernen: afbrydelse (interrupts), realisering af systemkald, drivere for ydre enheder, I/O planlægning og afvikling, hardwareunderstøttelse
- samtidighed/parallelisme: relativ tid, synkronisering, race-conditions, mutex, semaforer, monitor, fairness, baglåse, nødvendige og tilstrækkelige betingelser for baglås, strategier for baglåshåndtering, multi-core arkitekturer, parallel-programmering, teknikker og værktøjer til samtidig-/parallel-programmering, inter-tråd/-proces kommunikation

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- kunne redegøre præcist og ved brug af fagets terminologi og notation for opbygning, strukturering, funktionalitet og virkemåde af styresystemer
- kunne analysere simple, systemnære programmer der benytter sig af parallelitet og /eller samtidighed
- kunne anvende fagets teknikker til at sikre gensidig udelukkelse, fairness og fravær af baglås i simple samtidige/parallelle systemer

Kompetencer:

Den studerende skal, ved syntese af fagets begreber og teknikker, kunne udvikle systemnære simple programmer, der benytter sig af parallelitet og /eller samtidighed.

Den studerende skal kunne tilegne sig ny viden om styresystemer samt programmering af samtidige og parallelle systemer

Undervisningsform: Kursus

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Ekstern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

Titel: **Sprog og oversættere
(Languages and Compilers)**

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger: Programmeringserfaring svarende til projektmodulet på 3. semester samt kendskab til imperativ og objektorienteret programmering svarende til 1. og 2. semesters kurser i programmering

Mål: Viden:

Den studerende skal opnå viden om væsentlige principper i programmeringssprog, samt forståelse af teknikker til beskrivelse og oversættelse af sprog generelt, herunder:

- Abstraktionsprincippet, kontrol- og datastrukturer, blokstruktur og scopebegrebet, parametermekanismer og typeækvivalens

- Oversættelse, herunder leksikalsk, syntaktisk, og statisk semantisk analyse, samt kodegenerering
- Køretids-omgivelser, herunder lagerallokering samt strukturer til understøttelse af procedurer og funktioner

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- Kunne redegøre for de berørte teknikker og begreber inden for sprogdesign og oversætterkonstruktion ved brug af fagets terminologi og notation for beskrivelse og implementation af programmeringssprog
- Kunne redegøre for hvordan implementations teknikker influerer sprog design
- Kunne ræsonnere datalogisk om og med de berørte begreber og teknikker

Kompetencer:

Den studerende skal kunne beskrive, analysere og implementere programmeringssprog og skal kunne redegøre for de enkelte faser og sammenhængen mellem faserne i en oversætter

Undervisningsform: Kursus

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Ekstern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

5. semester, SW5

Titel: Indlejrede systemer
(Embedded Systems)

Omfang: 15 ECTS (Projektmodul)

Forudsætninger: Projektmodulerne på 1. - 4. semester i Bacheloruddannelsen for software, samt at ét af kursusmodulerne Tidstro software eller Maskinintelligens svarende til projektets fokus følges parallelt med projektarbejdet.

Formål: At den studerende opnår viden om principper for og færdigheder i udvikling af programmel til indlejrede systemer.

Begrundelse: Tekniske anlæg og apparater styres og reguleres i dag som hovedregel af programmer der er indlejret i dem. Det er derfor essentielt at softwareingeniører opnår forståelse og færdigheder i at udvikle sådant programmel under hensyntagen til de krav som det omgivende system stiller, samt de krav om høj pålidelighed som kræves for dem.

Mål: Viden:
Den studerende skal efter gennemført projektmodul kunne:

- benytte korrekte begreber (i både skrift og tale), notationer og symboler
- demonstrere kendskab til og overblik over grundlæggende teknikker fra tidstro systemer eller maskinintelligens som kan anvendes til konstruktioner af indlejrede systemer

Færdigheder:

Den studerende skal efter gennemført projektmodul kunne:

- anvende relevante og centrale teknikker inden for tidstro eller intelligente systemer i forbindelse med konstruktion af software til indlejrede systemer

Kompetence:

Den studerende skal efter gennemført projektmodulet være i stand til at:

- afgrænse og formulere problemer inden for indlejrede systemer
- vælge og begrunde valg af en platform til løsning af sådanne problemer
- udvikle applikationer til indlejrede platforme
- vurdere og begrunde valget af relevante teknikker og metoder til konstruktion af indlejrede systemer

Undervisningsform: Projektarbejde, hvor der kan arbejdes med et nærmere udvalgt emne inden for indlejrede systemer. Projektarbejdet kan omfatte såvel teoretisk analyse som praktisk implementation og eksperimenter, eller kombination deraf

Prøveform: Mundtlig prøve på baggrund af projektrapport

Bedømmelse: Intern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

Titel: **Maskinintelligens
(Machine Intelligence)**

Omfang: 5 ECTS (Valgfrit kursusmodul)

Forudsætninger: Kursus- og projektmodulerne på 1.- 4. semester af bacheloruddannelsen i software

Mål: Viden:
Den studerende skal opnå viden om følgende teorier og metoder:

- demonstrere kendskab til grundlæggende teknikker og metoder indenfor maskinintelligens
- anvende korrekt teknisk notation og terminologi i skrift såvel som tale

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- anvende grundlæggende teknikker præsenteret i kurset til løsning af en konkret problemstilling
- gøre rede for centrale principper og algoritmer præsenteret i kurset

Kompetencer:

Den studerende skal med udgangspunkt i en konkret problemstilling kunne vurdere og sammenligne forskellige teknikker og metoder inden for maskinintelligens

Undervisningsform: Forelæsning med tilhørende opgaveregning

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Intern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

Titel: **Tidstro software
(Real-time Systems)**

Omfang: 5 ECTS (Valgfrit kursusmodul)

Forudsætninger: Kursusmodulerne Computer arkitektur (2. semester) og Principper for styresystemer og parallelitet (4. semester)

Mål: Viden:

Den studerende skal opnå viden om følgende teorier og metoder:

- design: tasks, temporal scopes, filhåndteringsstrategier, mode, change, synkron og asynkron interaktion
- analyse: tidsplanlægning, svartidsanalyse, modellering, verifikation og validering, prioritetsprotokoller, hardwarebegrænsninger
- implementation: programmeringssprog med understøttelse for realtidsprogrammering, hardware-abstraktion og systemnær programmering, synkronisering, atomicitet, baglåse (deadlocks), fejlhåndtering, kommunikation

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- kunne redegøre præcist og ved brug af fagets terminologi og notation for overordnet design, analyse og implementation af simple tidstro softwaresystemer
- kunne anvende fagets teknikker til at afgøre mulighed for tidsplanlægning for en simpel realtids-applikation

Kompetencer:

Den studerende skal, ved syntese af fagets begreber og teknikker:

- kunne designe, analysere og implementere en simpel (indlejret) realtids-applikation
- kunne tilegne sig ny viden om design, analyse og implementation af realtids-systemer

Undervisningsform: Kursus

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Intern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

Titel: **Software Engineering
(Software Engineering)**

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger: Kursusmodulerne Systemudvikling og Design, implementation og vurdering af brugergrænseflader samt projektmodulerne SW3

Mål: Viden:
Den studerende skal opnå viden om førende paradigmer (f.eks. traditionelt og agilt) inden for professionel udvikling af programmer og systemer, samt teorier, metoder og teknikker som indgår i disse paradigmer (f.eks. procesmodeller, kravstyring, design, projektledelse, test, procesforbedring)

Færdigheder:
Den studerende skal opnå følgende færdigheder:

- Kunne redegøre præcist og ved brug af fagets terminologi for de udvalgte paradigmer, og kunne adskille og sammenligne disse
- Kunne redegøre præcist og ved brug af fagets terminologi for teorier, metoder og teknikker inden for paradigmerne og deres anvendelse i professionel udvikling af software intensive systemer

Kompetencer:
Den studerende skal kunne vælge, begrunde og anvende passende paradigmer, teorier, metoder og teknikker i deres egne udviklingskontekster

Undervisningsform: Kursus

Prøveform: Individuel mundtlig prøve

Bedømmelse: Ekstern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

Titel: **Beregnelighed og kompleksitet
(Computability and Complexity)**

Omfang: 5 ECTS

Forudsætninger: Kurset forudsætter kursusmodulerne Diskret Matematik (2. semester), Algoritmik og datastruktur (3. semester) og Syntaks og semantik (4. semester).

Mål: Viden:
Den studerende skal opnå viden om følgende teorier og metoder:

Beregnelighed:

- deterministiske og nondeterministiske Turing-maskiner; afgørbare og genkendelige sprog og deres egenskaber: Church-Turing-tesen
- acceptproblemet for Turing-maskiner; andre uafgørbare problemer for Turing-maskiner; reduktioner og deres egenskaber

Kompleksitetsteori:

- tidskompleksitet for deterministiske og nondeterministiske Turing-maskiner; tidskompleksitetsklasser; polynomielle reduktioner og deres anvendelser; NP-fuldstændighed; opfyldelighedsproblemet (SAT); øvrige NP-fuldstændige problemer
- pladskompleksitet for deterministiske og nondeterministiske Turing-maskiner; pladskompleksitetsklasser, forholdet mellem tids- og pladskompleksitet

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- kunne redegøre præcist og ved brug af fagets terminologi og notatin for vigtige resultater inden for teorierne for beregnelighed og beregningskompleksitet og for hvordan og i hvilket omfang disse resultater kan anvendes til at klassificere beregningsproblemer
- kunne gøre brug af de fornødne skriftlige færdigheder i disse sammenhænge

Kompetencer:

Den studerende skal kunne anvende begreber og teknikker fra teorierne for beregnelighed og beregningskompleksitet til analyse af beregningsproblemer

Undervisningsform: Kursus

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Ekstern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

6. semester, SW6

Titel: **Bachelorprojekt (Udvikling af komplekse softwaresystemer)**
BSc Project (Developing Complex Software Systems)

Omfang:	15 ECTS (Projektmodul)
Forudsætninger:	Projekt- og kursusmodulerne på 1.-5. semester samt at tre af de udbudte kursusmoduler på samme semester følges parallelt med projektarbejdet
Formål:	At den studerende opnår viden om og færdigheder i analyse, design, implementering og vurdering af komplekse softwaresystemer i et større udviklingsmiljø
Begrundelse:	Efter en fuldført uddannelse i softwareteknologi skal man kunne deltage i løsning af problemer ved at udvikle komplekse softwaresystemer. Typisk sker udvikling af sådanne komplekse systemer i et større udviklingsmiljø, hvor delprojekter skal koordineres med henblik på en fælles løsning. Den studerende skal derfor opnå forståelse af problemstillinger i forbindelse med udvikling af sådanne systemer, herunder analyse af organisatoriske omgivelser, samt design og implementering af en applikation som understøtter eller automatiserer en del heraf. At arbejde med at analysere, designe, programmere og afprøve en applikation med væsentlig funktionalitet, herunder både integration med database og brugergrænseflade som skal indgå i en kompleks organisatorisk omgivelse, sætter fokus på kravanalyse, kravstyring, ledelse mellem delprojekter, prototyping, programmering af store softwaresystemer, test, og avanceret anvendelse af softwareteknologier
Mål:	<u>Viden:</u> Den studerende skal efter gennemført projektmodul kunne: • Dokumentere kendskab til og overblik over centrale teknikker i arbejdet med at udvikle software, der løser realistiske problemer, herunder - kravanalyse - kravstyring - prototyping - databaser - usability - test og verifikation • Benytte korrekt fagterminologi <u>Færdigheder:</u> Den studerende skal efter gennemført projektmodul kunne: • analysere, designe, programmere, afprøve og teste applikationer som indgår i kompleks organisatorisk omgivelse • ræsonnere om og med de berørte begreber og teknikker • begrunde og vælge relevante læringsmodeller ud fra kendskab til de muligheder og begrænsninger, som er givet af fagområdets teorier og metoder <u>Kompetencer:</u> Den studerende skal efter gennemført projektmodul kunne: • afgrænse og gennemføre løsning af en del af et større softwareudviklingsproblem ved brug af relevante teknikker • analysere og vurdere læringsprocessen og den fremkomne løsning

Undervisningsform: Projektarbejde, der gennemføres som et multiprojekt, hvor der gennemføres en samlet udvikling af et komplekst softwaresystem på tværs af flere projektgrupper. Projektarbejdet skal indeholde:

- analyse af en organisatorisk omgivelse
- design af en applikation som understøtter en væsentlig del heraf
- udvikling af et program, som realiserer designet

Prøveform: Mundtlig prøve på baggrund af projektrapport

Bedømmelse: Ekstern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

Titel: **Avancerede algoritmer
(Advanced Algorithms)**

Omfang: 5 ECTS (Valgfrit kursusmodul)

Forudsætninger: Kursusmodulerne Algoritmik og datastruktur (3. semester), Principper for styresystemer og parallelitet (4. semester), og Beregnelighed og kompleksitet (5. semester)

Mål:

Viden:

Den studerende skal opnå viden om følgende teorier og metoder:

- algoritmedesign teknikker såsom del-og-hersk, grådige algoritmer, dynamisk programmering, back-tracking, forgren-og-begræns algoritmer og plane-sweep algoritmer
- algoritmeanalyse teknikker såsom rekursion, amortiseret analyse, analyse af forventet kompleksitet og eksperimenter med algoritmer
- en samling af kernealgoritmer og datastrukturer til løsning af en række problemer fra forskellige datalogiske områder: algoritmer til ekstern hukommelse, flere-trådede algoritmer, søgning i tekst, avanceret grafalgoritmer, heuristisk søgning og geometriske beregninger

Der vil desuden indgå et eller flere valgfri emner indenfor avancerede algoritmer, inklusiv, men ikke begrænset til: approksimative algoritmer, randomiserede algoritmer, lineær programmering og talteoretiske algoritmer såsom kryptosystemer

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- redegøre for principperne bag de vigtigste algoritme-design og - analyse teknikker
- udvælge og anvende algoritme-design og – analyse teknikker for en given problemstilling
- genkende en række problemer fra forskellige datalogiske områder og udvælge de mest passende algoritmer og datastrukturer for at løse dem

Kompetencer:

Den studerende skal, stillet over for en ikke-standard datalogisk problem kunne

- udvikle effektive algoritmer og datastrukturer til løsning af problemet
- analysere de udviklede algoritmer

Undervisningsform: Kursus

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Intern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

**Titel: Semantik og verifikation
(Semantics and Verification)**

Omfang: 5 ECTS (Valgfrit kursusmodul)

Forudsætninger: Kurset forudsætter kursusmodulerne Syntaks og semantik (4. semester) og Beregnelighed og kompleksitet (5. semester).

Mål: Viden:

Den studerende skal opnå viden om avancerede matematiske modeller til formel beskrivelse og verifikation af programmer, softwaresystemer og programmeringssprog med fokus på parallelle og kommunikerende systemer. Specielt skal den studerende opnå viden om:

- transitionssystemer
- procesalgebra, f.eks. CCS
- bisimulering
- Hennessy-Milner logik med rekursion
- Tarskis sætning om fikspunkter
- modeller og ræsonnementsmetoder for realtidssystemer, f. eks. Timed CCS og tidsautomater
- probabilistiske modeller og ræsonnementsteknikker, f. eks. probabilistiske proceskalkyler, ækvivalenser og logikker
- verifikationsteknikker for realtidsmodeller
- evt. andre verifikationsmetoder

Kursusmodulet kan derudover inddrage andre formelle modeller.

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- kunne redegøre præcist og ved brug af fagets terminologi og notation for vigtige teorier for beskrivelse og analyse af reaktive systemer
- kunne anvende verifikationsværktøjer, der er baseret på formelle modeller

- kunne gøre brug af de fornødne skriftlige færdigheder i disse sammenhænge

Kompetencer:

Den studerende skal kunne anvende formelle modeller og hermed forbundne verifikationsværktøjer til verifikation af softwaresystemer.

Undervisningsform: Kursus

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Intern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

Titel: **Databaser
(Databases)**

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger: Kursusmodulet Algoritmik og datastruktur (3. semester)

Mål: Viden:
Den studerende skal opnå indsigt inden for følgende emner i database systemer:

- den relationelle model og relationel algebra
- entity-relationship diagrammer (ERD)
- spørgesproget SQL
- logisk design af relationelle databaser (normal former)
- fysisk databasedesign (lager, filer, indeksering og hashing)
- forespørgselshåndtering og optimering
- transaktioner (transaktion begrebet, concurrency control og recovery)

En række mulige emner der også kan blive dækket i kurset afhængigt af underviserens og de studerendes evner og baggrund er:

- relationel calculus
- parallelle databaser
- distribuerede databaser
- advanced SQL (f.eks. triggers og stored procedures)
- entity-relationship diagrammer (ERD)

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- kunne forklare den relationelle model og anvende relationel algebra på et datasæt
- kunne konstruere et ERD for mindre, konkrete scenarier
- kunne lave et relationelle databasedesign der overholder anerkendte normal former

- kunne bruge SQL til at skabe og forespørge på en database
- kunne designe og anvende passende file- og indeksstrukturer på en given database
- kunne forklare en eksekveringsplan for en SQL forespørgsel og vurdere om planen er effektiv
- kunne forklare transaktionsbegrebet, og centrale emner indenfor concurrency control og recovery

Kompetencer:

Den studerende skal ved brug af de fundamentale begreber og teorier, der er fælles for de fleste databasesystemer, kunne anvende disse til i praksis at håndtere større datasæt

Undervisningsform: Kursus

Prøveform: Individuel mundtlig eller skriftlig prøve

Bedømmelse: Ekstern bedømmelse efter 7-trins-skala

Vurderingskriterier: Se Rammestudieordningen

Titel: **Videnskabsteori
(Theory of Science)**

Omfang: 5 ECTS (Kursusmodul)

Forudsætninger: Kursus- og projektmodulerne på 1. – 5. semester

Mål: Viden:

Den studerende skal opnå viden om følgende problemstillinger:

- datalogiens historie, grundlagsdiskussioner og relationer til andre videnskaber
- etiske spørgsmål inden for forskning i og anvendelse af datalogiens fagområder
- videnskabelig metode; empiriske og teoretiske tilgange
- teorier for erkendelse og modeldannelse. Kvantitative og kvalitative metoder og disses anvendelighed; design af eksperimenter, tilgange til anvendelse af statistik
- praksis for mundtlige og skriftlig kommunikation inden for datalogiens forskningsområder
- det videnskabelige samfund inden for datalogi; typer af videnskabelige publikationer, peer review-processen

Færdigheder:

Den studerende skal opnå følgende færdigheder:

- kunne redegøre for problemstillinger inden for datalogiens forskning og praksis ud fra en videnskabsteoretisk indsigt

- kunne benytte sig af gængse former for mundtlig og skriftlig kommunikation af videnskabelige art inden for datalogi
- kunne gøre brug af de fornødne skriftlige færdigheder i disse sammenhænge

Kompetencer:

Den studerende skal kunne

- analysere, vurdere og tage kritisk stilling til videnskabsteoretiske aspekter af forskning i og anvender af datalogiens fagområder
- reflektere over egen og andres mundtlige og skriftlige kommunikation af videnskabelig art inden for datalogi

Undervisningsform: Kursus

Prøveform: Aktiv deltagelse eller løbende evaluering

Bedømmelse: Intern bedømmelse, bestået/ikke-bestået

Vurderingskriterier: Se Rammestudieordningen

Kapitel 4: Ikrafttrædelse, overgangsregler og revision

Studieordningen er godkendt af dekanen for Det Teknisk - Naturvidenskabelige Fakultet og træder i kraft pr. 1. september 2015.

Studerende, der ønsker at færdiggøre deres studier efter den hidtidige studieordning, skal senest afslutte deres uddannelse ved sommereksamen 2017, idet der ikke efter dette tidspunkt udbydes eksamener efter den hidtidige studieordning.

I henhold til Rammestudieordningen for Det Teknisk - Naturvidenskabelige Fakultet ved Aalborg Universitet skal studieordningen tages op til revision senest 5 år efter dens ikrafttræden.

Kapitel 5: Andre regler

5.1 Regler om skriftlige opgaver, herunder bachelorprojektet

I bedømmelsen af samtlige skriftlige arbejder skal der ud over det faglige indhold, uanset hvilket sprog de er udarbejdet på, også lægges vægt på den studerendes stave- og formuleringsevne. Til grund for vurderingen af den sproglige præstation lægges ortografisk og grammatisk korrekthed samt stilistisk sikkerhed. Den sproglige præstation skal altid indgå som en selvstændig dimension i den samlede vurdering. Dog kan ingen prøve samlet vurderes til bestået alene på grund af en god sproglig præstation, ligesom en prøve normalt ikke kan vurderes til ikke bestået alene på grund af en ringe sproglig præstation.

Studienævnet kan i særlige tilfælde (f.eks. ordblindhed og andet sprog end dansk som modersmål) dispensere herfor.

Bachelorprojektet skal indeholde et resumé på engelsk¹. Hvis projektet er skrevet på engelsk, skal resumeet skrives på dansk². Resumeet skal være på mindst 1 og må højst være på 2 sider (indgår ikke i eventuelle fastsatte minimum- og maksimumsidetal pr. studerende). Resumeet indgår i helhedsvurderingen af projektet.

5.2 Regler om merit, herunder mulighed for valg af moduler, der indgår i en anden uddannelse ved et universitet i Danmark eller udlandet

Studienævnet kan i hvert enkelt tilfælde godkende, at beståede uddannelseselementer fra andre bacheloruddannelser træder i stedet for uddannelseselementer i denne uddannelse (merit). Studienævnet kan også godkende, at beståede uddannelseselementer fra en anden dansk eller udenlandsk uddannelse på samme niveau træder i stedet for uddannelseselementer efter denne studieordning. Afgørelser om merit træffes af studienævnet på baggrund af en faglig vurdering. For regler om merit se Rammestudieordningen.

5.3 Regler omkring forløb og afslutning af bacheloruddannelsen

Inden udgangen af første studieår på bacheloruddannelsen skal den studerende, for at kunne fortsætte uddannelsen, deltage i alle prøver på første studieår. Første studieår skal være bestået senest inden udgangen af andet studieår efter studiestart, for at den studerende kan fortsætte sin bacheloruddannelse.

Der kan dog i særlige tilfælde dispenseres fra ovenstående.

5.4 Afslutning af bacheloruddannelsen

Bacheloruddannelsen skal være afsluttet senest seks år efter, den er påbegyndt.

5.5 Særligt projektforsløb

Den studerende kan på 3., 4. eller 5. semester, efter ansøgning, sammensætte et uddannelsesforsløb, hvor projektarbejdet erstattes af andre studieaktiviteter jf. Rammestudieordningens afsnit 9.3.1.

5.6 Eksamensregler

¹ Eller et andet et fremmedsprog (fransk, spansk eller tysk) efter studienævnets godkendelse

² Studienævnet kan dispensere herfra

Eksamensreglerne fremgår af eksamensordningen, der er offentliggjort på Det Teknisk – Naturvidenskabelige Fakultets hjemmeside.

5.7 Dispensation

Studienævnet kan, når der foreligger usædvanlige forhold, dispensere fra de dele af studieordningens bestemmelser, der ikke er fastsat ved lov eller bekendtgørelse. Dispensation vedrørende eksamen gælder for den først kommende eksamen.

5.8 Regler og krav om læsning af tekster på fremmedsprog og angivelse af hvilket kendskab til fremmedsproget(ene) dette forudsætter

Det forudsættes, at den studerende kan læse akademiske tekster på moderne dansk, norsk, svensk og engelsk samt anvende opslagsværker mv. på andre europæiske sprog.

5.9 Uddybende information

Gældende version af studieordningen er offentliggjort på studienævnets hjemmeside, herunder mere udførlige oplysninger om uddannelsen, herunder om eksamen.