

# DB1100 Databaser 1

## Faglig innhold

### Kort om emnet

Etter fullført Databaser 1 skal man kunne forklare hva en relasjonsdatabase er, hva den kan brukes til og hvordan den skiller seg fra andre former for persistent lagring. Man skal kunne modellere og strukturere data for et domene. Man skal kunne opprette tabeller, legge inn ulike typer data i disse, kople dem sammen og hente ut rapporter ved hjelp av SQL spørringer. Man skal kunne forklare og anvende prinsippene for god design (normalisering, nøkkelbruk).

### Kunnskap

- Lære grunnleggende C# syntaks, samt vite på hvilke måter språket skiller seg fra Java.
- Forstå hvordan prinsippene bak objektorientert programmering kan brukes i C# med Unity3d.
- Forstå hensikten med en "game loop" og hvordan denne er nødvendig for spill.
- Kjenne til Unitys event functions og bruksområdene for disse.
- Forstå hensikten med local og world koordinater.
- Kjenne til hvordan rotasjon, translasjon og skalering benyttes til å manipulere objekter.
- Kjenne til hvordan kollisjoner implementeres i Unity.
- Kjenne til hvordan mus, tastatur og gamepad benyttes for input.

### Ferdigheter

- Kunne benytte programmeringspråket C#.
- Debugge, lokalisere og rette feil i programmer.
- Beherske Unity utviklingsmiljøet, kunne opprette prosjekter og publisere de til forskjellige plattformer.
- Importere og anvende 3D-modeller, animasjoner, teksturer, lyder, "prefabs" og andre assets.
- Legge til spillmekanikk ved hjelp av programmering.

### Generell Kompetanse

- Planlegge, utvikle og publisere enkle spill.
- Implementere og utvide 3D spillomgivelser.

## Eventuelle forkunnskapskrav

### Generell studiekompetanse

### Pensum

| Tittel   | ISBN              |
|--|-------------------|
| Database Systems. A Practical Approach to Design, Implementation, and Management | 978-0-321-52306-8 |

## Undervisningsform

Forelesninger, øvinger og selvstudium.

## Annen informasjon som er relevant.

### Vitnemålstekst

Etter å ha fullført emnet skal man kunne forklare hva en relasjonsdatabase er, hva den kan brukes til

og hvordan den skiller seg fra andre former for persistent lagring. Man skal kunne modellere og strukturere data for et domene. Man skal kunne opprette tabeller, legge inn ulike typer data i disse, kople dem sammen og hente ut rapporter ved hjelp av SQL spørringer. Man skal kunne forklare og anvende prinsippene for god design (normalisering, nøkkelbruk), og utføre grunnleggende administrative oppgaver.

## PG1100 Programmering 1

### Faglig innhold

#### Kort om emnet

I utviklingen av applikasjoner er programmering en meget sentral disiplin, siden det er dette som benyttes til å realisere /lage kjørbare systemer på en datamaskin. Denne disiplinen trekkes direkte eller indirekte inn i en rekke andre fag ved IT-utdanningen. Emnet skal gi en første innføring i programmering og utvikling av algoritmer for å løse ulike problemer.

#### Kunnskaper

Etter å ha fullført emnet skal studenten

- Vite hva et programmeringsspråk er, og hvordan slike skiller seg fra naturlige språk
- Vite forskjellen på interpreterte og kompilerte språk
- Kjenne til forskjellene på hendelsesdrevne og rent sekvensielle programmer

#### Ferdigheter

Etter å ha fullført emnet skal studenten kunne

- Bruke et API
- Lage enkle programmer som bl.a. gjør bruk av:
  - Variabler
  - Tilordning
  - Ulike datatyper
  - Aritmetiske og logiske uttrykk
  - Lage enkle programmer som anvender kontrollstrukturene
  - Valg
- LøkkeLage enkle programmer med fokus på utvikling av algoritmer
- Lage enkle programmer som skriver til og leser fra tekstfiler
- Deklarere og bruke statiske tabeller (arrays)
- Beherske et enkelt integrert utviklingsverktøy (IDE)
- Kunne skrive og redigere kildekode med dette verktøyet
- Kunne kjøre programmer med dette verktøyet
- Kunne bruke de grunnleggende elementene i et programmeringsspråk til å lage programmer
- Kunne tolke og forholde seg til feilmeldinger og rette kildekode ut fra disse

#### Generell kompetanse

Etter å ha fullført emnet skal studenten kunne

- Vurdere passende strategi for løsning av et problem
- Diskutere problemløsningsmetoder med medstudenter
- Gjøre rede for tankegang for egne løsninger

#### Eventuelle forkunnskapskrav

Emnet krever ingen konkrete faglige forutsetninger utover generell studiekompetanse og god kjennskap til bruk av en PC. Programmering krever en høy grad av nøyaktighet, tålmodighet, utholdenhet og evne til logisk tenking.

## Pensum

| Tittel                 | ISBN           |
|------------------------|----------------|
| Building Java Programs | 978-0133360905 |

## Undervisningsform

Forelesninger, øvinger med veileder, innleveringer og skriftlig vurdering.

## Annen informasjon som er relevant.

### Vitnemålstekst

Emnet har gitt en introduksjon i programmering med vekt på utvikling av algoritmer for å løse enkle problemer. Studenten kan deklare og anvende ulike datatyper og gjøre bruk av kontrollstrukturene valg og løkke. Studenten kan også gjøre bruk av tabeller (arrays) etter behov.

## PJ1100 Kreativt webprosjekt

### Faglig innhold

#### Kort om emnet

Etter å ha fullført emnet skal studenten gjennom teamarbeid kunne utføre en kreativ prosess rettet mot design og implementering av en IT-løsning. Studenten skal kunne skille mellom konsept, innhold, struktur og presentasjon for et nettsted, og utføre enkel oppmarkering med HTML og CSS av en statisk nettløsning.

### Kunnskaper

Etter å ha fullført emnet skal studenten kunne:

- Forklare hva kreativitet er
- Forklare hva HTML og CSS er
- Forklare konseptet semantikk i forhold til HTML
- Forklare oppbygningen av en HTML tag (med attributter)
- Forklare oppbygningen av en CSS-regel

### Ferdigheter

Etter å ha fullført emnet skal studenten kunne:

- Bruke kreativitet i utvikling av IT-systemer
- Bruke CPS (Creative Problem Solving) som metode for problemløsning
- Bruke ulike teknikker i en kreativ prosess, ulike typer brainstorming, idéskrivning, og bruk av sansene
- Benytte html for å legge ut tekst, bilder og lenker
- Forstå og benytte forskjellige typer CSS-velgere

## Eventuelle forkunnskapskrav

Ingen

## Pensum

| Tittel   | ISBN          |
|--|---------------|
| Praktisk nytenkning: systematisk og kreativ problemløsning | 9788244300001 |

## Undervisningsform

Emnet gjennomføres med forelesning og øvinger, og avsluttes med to uker prosjekt hvor studentene løser en gitt caseoppgave i grupper med avsluttende presentasjon.

## Annen informasjon som er relevant.

### Vitnemålstekst

Emnet har gitt studenten innføring i teamarbeid for å kunne utføre en kreativ prosess rettet mot design og implementering av en IT-løsning ved hjelp av HTML og CSS.

## TK1100 Digital teknologi

### Faglig innhold

#### Kort om emnet

For å kunne benytte en datamaskin på en effektiv måte må man vite hvordan informasjon kodes digitalt, samt hvordan den lagres, prosesseres og overføres av og mellom maskinvare og programvare. Ved å arbeide med emnet skal studenten lære seg å analysere datasystemer i ulike abstraksjonslag fra bit-nivå, via digitale kretser og maskinvarekomponenter (CPU, minne, busser og ulikt I/O-utstyr), data vs instruksjoner, operativsystem, applikasjoner og nettverkskommunikasjon. De skal kunne forklare hvordan man med binærtall kan representere ulike former for informasjon. De skal erverve seg begrepsapparatet som trengs for å vurdere ulik maskin- og programvare opp mot hverandre. De skal kunne benytte modeller for funksjonell lagdeling i systemer, samt prosedyrer og verktøy til å forklare virkemåte og derigjennom kunne utføre effektiv feilsøking av enkeltmaskiner og nettverkskommunikasjon.

#### Kunnskaper

Etter å ha fullført emnet skal studenten kunne:

- Beskrive den historiske utviklingen som ledet frem til dagens datamaskiner og nettverk
- Definere og benytte de vanligste begrepene som benyttes i beskrivelse og analyse av datasystemer og nettverk
- Prinsippene og metodene for digital koding av ulike former informasjon og hvilke muligheter og begrensninger som er forbundet med disse
- Beskrive arkitekturen og organiseringen av datasystemer og nettverk
- Forklare rollen til et operativsystem og dekomponere funksjonaliteten i hovedelementene prosess/tråd-administrasjon (scheduling + samtidshåndtering), minne-administrasjon (virtuelt minne mm), fil-administrasjon, I/O-administrasjon
- Benytte TCP/IP-modellen til å forklare og analysere datakommunikasjon gjennom Internett

#### Ferdigheter

Etter å ha fullført emnet skal studenten kunne:

- Identifisere ulike kodingsformater for tekst, tall, lyd og bilder; kjenne virkemåten til disse og benytte dette i feilsøking
- Med utgangspunkt i kjennskap til virkemåten til hovedkort, CPU, minne, I/O-kontrollere, busser og persistente lagringsmedia; identifisere de viktigste metrikker og benchmarks for ulike typer utstyr og identifisere mulige feilkilder og hvordan disse vil ytre seg
- Benytte skall-kommandoer, overvåkings- og konfigurasjons- verktøy i Windows og/eller OSX/Linux
- Benytte TCP/IP-modellen og kjennskap til tilhørende portokoller, til å analysere den fullstendige gangen i tilkopling til et LAN og nedlastingen av f.eks. en webisde

## Generell kompetanse

Typiske oppgaver en student skal være i stand til å løse etter å ha fullført emnet er:

- Gi en anbefaling ved innkjøp av datamaskin ut fra brukers behov og ønsker med bakgrunn i en korrekt og systematisk forståelse av tekniske spesifikasjoner for utstyret
- Gjennomføre systematisk feilsøking og reparasjon av typiske problemer i hardware og software
- Gi en begrunnet anbefaling ved valg av ISP og nettprodukt/abonnementstype, samt lokalt nettverksutstyr ut fra overslag på behov for bitrate og tjenestekvalitet
- Benytte standard verktøy for å finne feil på og rette opp nettverksforbindelser

## Eventuelle forkunnskapskrav

Generell studiekompetanse

## Pensum

www-ressurser og kompendium som publiseres på emnesiden i It's Learning

## Undervisningsform

Forelesninger, elektroniske tester og individuelle øvingsoppgaver med veiledning.

## Annen informasjon som er relevant.

### Vitnemålstekst

Etter å ha fullført emnet skal studenten kunne forklare hva slags maskinvare som inngår i en moderne datamaskin, hvordan komponentene samvirker og hvordan ulike typer informasjon kodes digitalt. Man skal kunne konfigurere og kjenne til oppbyggingen av et operativsystem. Man skal kunne redegjøre for prinsippene for datakommunikasjon i nettverk og protokollene i TCP/IP-stacken. Man skal kunne benytte vanlige verktøy for å feilsøke og rette opp feilsituasjoner på maskin, i operativsystem og i nettverk.

## TK2100 Informasjonssikkerhet

### Faglig innhold

#### Kort om emnet

Trusselbildet for en datamaskinbruker er i dag preget av angrep fra datakriminelle som er ute etter direkte økonomisk gevinst, eller å overta enkeltmaskiner for å benytte disse videre til kriminell virksomhet. Bevissthet om de ulike truslene som finnes i Internett er forutsetningen for å treffe riktige tiltak. Etter å ha fullført emnet skal en student være i stand til å analysere trusselbildet og foreta egnede sikringstiltak på egen maskin, i eget hjemmenettverk og gi begrunnede råd i forhold til oppsett og teknologivalg for websteder. Man skal også ha oversikt over hvilke lover og forskrifter som gjelder for bruk av datamaskiner til lagring, prosessering og formidling av data, her under personvern og opphavsrett

#### Kunnskaper

Etter å ha fullført emnet skal studenten kunne:

- Forklare hva som menes med informasjonssikkerhet.
- Beskrive de ulike nivåene i informasjonssikkerhet med tilhørende trusselbilde og (mulige) tiltak.
- Beskrive og forklare hvilke sikringstiltak som er tilgjengelig i ulike teknologier
- Vurdere sikkerhetstiltak i forhold til gitte situasjoner

## Ferdigheter

Etter å ha fullført emnet skal studenten kunne:

- Utføre en risikoanalyse i forhold til en bestemt teknologisk løsning og foreslå tilpassede tiltak
- Planlegge og gjennomføre (enkle) sikkerhetstester av bestemte teknologiske løsninger ved hjelp av standardverktøy

## Generell kompetanse

Etter å ha fullført emnet skal studenten kunne:

- Beskrive og følge lover og etiske normer som gjelder mhp opphavsrett og informasjonssikkerhet i Norge
- Reflektere over sikkerhetsaspekter ved egen nett-adferd.

## Eventuelle forkunnskapskrav

TK1100, DB1100, PG1100 og PJ1100

## Pensum

| Tittel                            | ISBN              |
|-----------------------------------|-------------------|
| Introduction to Computer Security | 978-0-321-70201-2 |

## Undervisningsform

Forelesninger og øvinger

## Annen informasjon som er relevant.

### Vitnemålstekst

Etter å ha fullført emnet skal studenten kjenne til og kunne beskrive tekniske, lovmessige og holdningsmessige sider ved informasjonssikkerhet for privatpersoner og virksomheter i forhold til bruk av Internett. De skal kunne utføre en systematisk risikovurdering av egen konfigurasjon og bruk av datamaskin på Internett, og kunne gi begrunnede råd til andre. De skal ha tilegnet seg fagterminologien som benyttes innenfor datasikkerhetsfeltet.

## PG2100 Programmering 2

### Faglig innhold

#### Kort om emnet

Emnet skal gi en første innføring i objektorientert programmering. Flere sentrale mekanismer i objektorientert programmering vil bli gjennomgått, samt eksempler på samspill mellom objekter av noen få klasser. Modellering i UML av små systemer (klassediagrammer og sekvensdiagrammer) blir også brukt.

Emnet skal gi en enkel innføring i programmering av grafiske grensesnitt (GUI) med hendelseshåndtering. Emnet gir en introduksjon av en standard Collection-klasse (ArrayList).

### Kunnskaper

Etter å ha fullført emnet skal studenten:

- Vite hva arv er
- Vite hva polymorfi er
- Vite om likheter/forskjeller mellom abstrakte klasser og interfaces
- Vite hvordan mekanismen for hendelseshåndtering fungerer

## Ferdigheter

Etter å ha fullført emnet skal studenten kunne:

- Definere klasser, og kunne beherske bruk av objekter
- Beherske ulike UML-diagramtyper slik som enkle klassesdiagrammer og sekvensdiagrammer
- Beherske bruk av objekter og få objekter til å samarbeide
- Kunne definere spesialiseringer av klasser gjennom arv
- Beherske bruk av abstrakte klasser og interface ('kontrakter')
- Anvende arv og interfaces gjennom kode med polymorfisk oppbygning
- Programmere enkle grafiske grensesnitt
- Bruke ArrayList-klassen til å opprette og bruke en dynamisk datastruktur

## Generell kompetanse

Etter å ha fullført emnet skal studenten kunne:

- Beherske et enkelt integrert utviklingsverktøy (IDE)
- Skrive og redigere kildekode med dette verktøyet
- Kjøre programmer med dette verktøyet
- Bruke de grunnleggende elementene i dette språket til å lage programmer
- Bruke en debugger for å finne feil

## Eventuelle forkunnskapskrav

PG1100

## Pensum

| Tittel                 | ISBN           |
|------------------------|----------------|
| Building Java Programs | 978-0133360905 |

## Undervisningsform

Foresninger, øvinger, innleveringer og skriftlig vurdering.

## Annen informasjon som er relevant.

### Vitnemålstekst

Emnet har gitt en innføring i objektorientert programmering. Studenten kan definere og anvende spesialiseringer av klasser gjennom arv/interface/polymorfi og mestrer programmering av enkle grafiske brukergrensesnitt (GUI) med tilhørende enkel hendelsehåndtering. I tillegg har studenten lært å bruke en standard Collection-klasse til å implementere en dynamisk datastruktur.

## PJ2100 Iterativt webprosjekt

### Faglig innhold

#### Kort om emnet

Etter å ha fullført emnet skal studenten kunne kjenne til ulike metoder for utvikling av programvare, og kunne gjennomføre et prosjekt med iterativ utvikling. Studenten skal kunne utforme en nettløsning i henhold til grunnleggende brukbarhetskrav med enkel databasefunksjonalitet.

### Kunnskaper

Etter å ha fullført emnet skal studenten kunne:

- Kjenne til ulike iterative og kundefokuserte utviklingsprosesser

- Ha kunnskap om roller i iterative utviklingsprosesser
- Kjenne til grunnleggende webdesign, grafisk grensesnitt og struktur
- Ha kunnskap om testing av brukbarhetskrav

### Ferdigheter

Etter å ha fullført emnet skal studenten kunne:

- Kunne anvende teknikk for dokumentasjon av brukerkrav
- Kunne gjennomføre grunnleggende webdesign, med vekt på brukbarhetskrav
- Kunne bruke et versjonshåndteringssystem for administrasjon av prosjektdokumentasjon
- Kunne gjennomføre et mindre prosjekt med iterativ struktur med en intern oppdragsgiver

### Generell kompetanse

Etter å ha fullført emnet skal studenten kunne:

- Kunne dokumentere og vurdere prosjektresultatet etter gitte kriterier
- Kunne dokumentere gjennomført prosjektprosess

### Eventuelle forkunnskapskrav

Grunnleggende database- og programmeringskunnskaper

### Pensum

| Tittel  | ISBN           |
|---|----------------|
| Don't Make Me Think: A Common Sense Approach to Web Usability   | 978-0321344755 |
| Microsoft Solutions Framework Essentials  | 978-0735623538 |
| Software Development Methods and Usability. Perspectives from a Survey in the Software Industry in Norway |                |

### Undervisningsform

Emnet gjennomføres med forelesninger og øvinger, og avsluttes med to uker prosjekt. Studentene løser en gitt caseoppgave i grupper med avsluttende presentasjon.

### Annen informasjon som er relevant.

#### Vitnemålstekst

Emnet gir en innføring i prosjektarbeid med iterativ utvikling og grunnleggende webdesign.

## PG3300 Programvarearkitektur

### Faglig innhold

#### Kort om emnet

Kurset skal gjøre studentene i stand til å designe og videreutvikle større programvaresystemer i tråd med kjente teknikker for modellering, testing og implementasjon.

#### Kunnskap:

- Vite bakgrunn og innhold for UML standarden.
- Kjenne til konsepter for versjonskontroll.
- Vite hva Unit Testing er.
- Forklare prinsippene for test-driven development (TDD).
- Vite hva design patterns er.
- Kjenne til og kunne identifisere en del viktige design patterns når de ser de i kode.
- Vite hva refactoring er.



- Vite hva multi-threading er.
- Forstå hvordan locking/lås kan brukes for å kode trådsikkert
- Forklare hvordan action listeners kan brukes for event handling.

#### Ferdigheter:

- Ha oversikt over og kunne benytte den grunnleggende syntaksen i programmeringspråket C# og vite på hvilke måter denne skiller seg fra Java.
- Beherske UML-diagrammene: use case, klassediagram, komponent diagram og sekvensdiagram.
- Bruke UML til å designe programarkitektur.
- Implementere følgende patterns: MVC & MVP, singleton, factory, builder, flyweight, composition, decorator.
- Kunne anvende design guidelines, som layers og GRASP prinsippene (med fokus på følgende for sistnevnte: controller, information expert, low coupling, high cohesion).
- Kunne gjennomføre refactoring på eksisterende kode.
- Kunne implementere applikasjoner som benytter flere tråder.
- Beherske et integrert utviklingsverktøy (IDE).
- Skrive og redigere kildekode med dette verktøyet.

#### Generell Kompetanse:

- Reflektere over multi-threading og dets bruksområder.
- Samarbeide med andre programmerere på en effektiv måte for å utvikle gode programmer.
- Vurdere kvalitet på eksisterende programmer og foreslå strukturelle forbedringer.

#### Eventuelle forkunnskapskrav

Erfaring med grunnleggende objektorientert programmering.

#### Pensum

| Tittel                    | ISBN           |
|---------------------------|----------------|
| Applying UML and Patterns | 978-0131489066 |

#### Undervisningsform

Forelesninger, øvinger og egenstudier.

#### Annen informasjon som er relevant.

#### Vitnemålstekst

Kurset skal gjøre studentene i stand til å designe og videreutvikle større programvaresystemer i tråd med kjente teknikker for modellering, testing og implementasjon.

## PG3400 Programmering i C for Linux

### Faglig innhold

#### Kort om emnet

Emnet gir innføring i programmeringspråket C og hvordan dette kan brukes til å interagere direkte med operativsystemet. Det skal også gi forståelse av operativsystemet Linux og kunnskap om hvordan dette virker.

#### Kunnskaper

Etter å ha fullført emnet skal studenten:

- Kjenne til operativsystemene Unix og Linux:
  - Bakgrunn og historie
  - Overordnet arkitektur og oppbygning

- Kjenne til utviklingen av og bruksområdet for programmeringspråket «C»
- Forstå hva «virtualisering» er
- Kjenne til byggeprosessen i C inkludert:
  - Compile
  - Link
- Forklare konseptet «everything is a file» og kjenne til eksempler inkludert:
  - Devices
  - Procs
- Forstå hva et «filsystem» er og hva det brukes til
- Forstå konseptene bak brukere og permissions i Linux
- Kjenne til forskjellen på userspace og kernelspace og hvordan bruke systemkall og signaler
- Kjenne til minneområdene «stack» og «heap» og forklare forskjellen

### Ferdigheter

Etter å ha fullført emnet skal studenten:

- Bruke grunnleggende konstruksjoner i programmeringspråket «C» inkludert:
  - main, basic types, printf, scanf, strings, #include, enum, typedef, struct, union, command line parameters, macros, varargs
  - Statisk minnehåndtering
- Utvikle programmer under paradigmet «strukturet programmering»
- Anvende dynamisk minnehåndtering inkludert:
  - Malloc, free, pointers, memcpy, sprintf, arrays, sizeof
- Lese og skrive enkeltbits og grupper av bits in en større datastruktur.
- Generere kjørbare filer fra C-kode
  - Makefile
  - Static libraries
- Implementere lesing og skriving til fil
  - r/w modes
  - Seek
  - Buffer size
- Operere kommandolinjen i linux for å utføre viktige operasjoner inkludert:
  - ls, cd, mkdir, rm, cp, chmod, chown, pwd, ps, kill, man, wget
  - Pipes, redirection
  - Starte programmer
  - Mounting av filsystemer

### Generell kompetanse

Etter å ha fullført emnet skal studenten kunne:

- Vurdere hvilke situasjoner det er fornuftig å bruke programmeringspråket «C»
- Anvende «C» for å løse systemprogrammeringsoppgaver
- Forholde seg til manuell minnehåndtering
- Forholde seg til linux som operativsystem

### Eventuelle forkunnskapskrav

Grunnleggende programmeringskunnskap (PG2100 Programmering 2 eller tilsvarende) samt grunnleggende digital teknologi (TK1100 Digital teknologi eller tilsvarende).

### Pensum

|                  |               |
|------------------|---------------|
| Tittel           | ISBN          |
| Programming in C | 9780672326660 |

## Undervisningsform

Kurset gjennomføres med 12 forelesninger a ca 2 timer og 12 lærerstyrte øvinger a ca 2 timer. Øvingene er ikke obligatoriske, men for å oppnå kompetansemålene forventes det at studentene fullfører øvingsoppgavene på egenhånd der øvingstimene ikke strekker til.

## Annen informasjon som er relevant.

### Vitnemålstekst

Etter å ha gjennomført emnet skal studentene kunne bruke operativsystemet Linux. De skal også være i stand til å programmere i programmeringspråket «C».

## RF3100 Matematikk og fysikk

### Faglig innhold

#### Kort om emnet

Moderne dataspill er basert på bruk av to- og tredimensjonal grafikk. En vesentlig del av spillene er å få fram bevegelser som likner dem vi kjenner fra virkeligheten. Det er derfor viktig for en spillprogrammerer å forstå hvilke fysiske lover som bestemmer legemers bevegelse, og å kunne beskrive disse bevegelsene ved hjelp av matematikk og geometri.

#### Kunnskap

Etter å ha fullført emnet skal studenten

- Kjenne til ulike koordinatsystemer,
- Forstå funksjonsbegrepet,
- Kjenne til vinkelmålet radianer,
- Kjenne definisjonene til de trigonometriske funksjonene sinus, cosinus og tangens,
- Kjenne til vektorer og matriser,
- Forstå begrepene gjennomsnittlig vekst, momentan vekst og derivasjon,
- Kjenne til Newtons tre lover,
- Kjenne til størrelsene energi og arbeid,
- Kjenne til loven om bevaring av mekanisk energi,
- Kjenne til størrelsene bevegelsesmengde og impuls,
- Kjenne til hvordan man beskriver rotasjonsbevegelse.

#### Ferdigheter

Etter å ha fullført emnet skal studenten kunne

- Regne med kartesiske koordinater og polarkoordinater i planet og i rommet,
- Tolke og drøfte grafen til en funksjon av en variabel, kunne sette opp og løse nødvendige likninger for å regne ut skjæringspunktene mellom grafer,
- Beregne kollisjonssirkler og -sfærer,
- Bruke trigonometri til nødvendige beregninger,
- Beskrive enkle periodiske fenomener ved hjelp av trigonometriske funksjoner,
- Behandle vektorer geometrisk og på koordinatform,
- Gjøre ulike operasjoner på matriser, og bruke matriser til å gjøre transformasjoner,
- Bruke bevegelseslikningene for translasjon til å bestemme forflytning, fart og akselerasjon,
- Definere og beregne vinkelbevegelse, vinkel- og tangentiell fart og akselerasjon,
- Bruke bevegelseslikningene for rotasjon til å bestemme størrelsene vinkelfart og vinkelakselerasjon,
- Bruke Newtons lover for translatorisk bevegelse og rotasjonsbevegelse,
- Beregne arbeid, kinetisk og potensiell energi og bruke energibevaringsloven for mekanisk energi,
- Bruke impuls-satsen til å beregne utfallet av kollisjoner,

- Definere og beregne dreiemoment.

### Generell kompetanse

Etter å ha fullført emnet skal studenten kunne

- Forstå grunnleggende lover i mekanikken slik at de blir i stand til å lage realistiske modeller av hvordan legemer beveger seg under påvirkning av krefter,
- Bruke matematikk til å beskrive legemers bevegelse,
- Representere punkters og legemers posisjon i rommet ved hjelp av matematikk.

### Eventuelle forkunnskapskrav

Generell studiekompetanse og R1 eller tilsvarende.

### Pensum

| Tittel   | ISBN          |
|--|---------------|
| 3D Math primer for graphics and game development | 9781568817231 |

### Undervisningsform

Forelesninger, øvinger med veileder og skriftlige vurderinger.

### Annen informasjon som er relevant.

#### Vitnemålstekst

Emnet har gitt studenten innsikt i de grunnleggende fysiske lovene som beskriver legemers bevegelse i to og tre dimensjoner. Emnet gir en innføring i de matematiske metodene som er nødvendige for å beskrive og beregne slike bevegelser. Temaer som tas opp er Newtons lover, spinnsetningen, bevegelsesmengde, skjæring mellom grafer, periodiske funksjoner, vektorer, matriser og transformasjoner.

## PJ3100 Prosjekt; Software Engineering

### Faglig innhold

#### Kort om emnet

Hensikten med dette emnet er å gi studenten en dypere erfaring i å mestre helheten i et større systemutviklingsprosjekt, med vekt på anvendelse av Scrum. Studenten skal gjennomføre et omfattende prosjektcase, og vil få trening å bruke moderne teknikker og verktøy.

#### Kunnskaper

Etter å ha fullført emnet skal studenten:

- Kjenne til grunnleggende utviklingsmetoder for programvare
- Kjenne til ulike smidige metoder og deres fordeler og ulemper
- Kunne beskrive hensikten med og strukturen i Scrum
- Kjenne til rammeverk for analyse av teamprosessen
- Kjenne til hvordan man arbeider effektivt i grupper

#### Ferdigheter

Etter å ha fullført emnet skal studenten:

- Kunne analysere gruppeprosessen i et team
- Kunne anvende ulike motivasjonsteorier i praksis under arbeid i team
- Kunne planlegge et prosjekt og utvikle en prosjektplan i tråd med Scrum

- Kunne designe/konfigurere/programmere og teste en større applikasjon som tilfredsstillere i et case, i et sammensatt teknologi-miljø
- Kunne anvende et prosjektstyringsverktøy til planlegging og oppfølging av prosjektet
- Kunne bruke et versjonshåndteringssystem

### Generell kompetanse

Etter å ha fullført emnet skal studenten:

- Kunne samarbeide i en prosjektgruppe om planlegging og gjennomføring av oppgaven
- Kunne dokumentere resultat og prosjektprosess i en sluttrapport
- Kunne reflektere over erfaringer i et prosjekt i forhold til utviklingsprosess og sluttresultat

### Eventuelle forkunnskapskrav

Emnet bygger på kunnskaper fra 1.klasse. Prosjektet forutsetter normal progresjon gjennom 2.klasse, da det integrerer kunnskaper og ferdigheter fra emner i de respektive studieprogrammene.

### Pensum

| Tittel                                     | ISBN              |
|--|-------------------|
| Agile Project Management with Scrum        | 978-0735619937    |
| Project management for Information Systems | 978-0-132-06858-1 |

### Undervisningsform

Prosjektet går over to semestre, knyttet til prosjektuker (seks sprinter). Prosjektet gjennomføres med prosjektgrupper på 4-5 studenter. Alle grupper får en egen veileder. Det vil være forelesninger i tilknytning til ulike kompetansemål og prosjektukene.

### Annen informasjon som er relevant.

#### Vitnemålstekst

Emnet gir en praktisk erfaring i et større systemutviklingsprosjekt med bruk av Scrum, i en gruppe på fire - fem studenter over to semestre.

## PG4200 Algoritmer og datastrukturer

### Faglig innhold

#### Kort om emnet

Emnet skal gi innsikt i algoritmer og datastrukturer som er sentrale i arbeidet med implementasjon og design av effektive datasystemer. Det legges vekt på en asymptotisk analyse av worst-case ressursbruk, samt sentrale algoritmer og datastrukturer knyttet til søk og sortering. Emnet tar også for seg enkelte graf-algoritmer.

#### Kunnskapsmål

Studentene skal kjenne til følgende:

- Sentrale abstrakte datatyper:
  - Lister, køer, stakker,
  - Mengder (sets, collections), Avbildninger (maps)
  - Trær og grafer.
- Egenskapene til sentrale datastrukturer:
  - Tabeller (arrays).
  - Lenkede lister.
  - Binære trær, søketrær, balanserte søketrær, B-trær, AVL-trær, Red-Black-trær.
  - Hash-tabeller og Hash-avbildninger.

- Grafer implementert ved naboskapslister og -matriser.
- Sentrale søkealgoritmer:
  - Linært søk, Binært søk.
  - Søk i binære søketrær.
  - Søk i Hash-tabeller.
- Sorteringsalgoritmer:
  - Insertion sort, Selection sort, Bubble sort,
  - Quicksort, Merge sort, Heap sort,
  - Radix sort.
- Enkelte graf-algoritmer, inkludert
  - Traversering av grafer,
  - Dijkstras algoritme, A\*-algoritmen,
  - Prims algoritme.
- Oppbygning, virkemåte og bruk av rekursive funksjoner, inkludert
  - Rekursiv traversering av trær og grafer,
  - Rekursiv søk og sortering,
  - Backtracking.
- Grunnleggende begreper innen teorien for beregningskompleksitet, inkludert
  - O-notasjon,
  - NP-kompletthet og reduksjon,
  - Den handelsreisendes problem (Travelling salesman problem),
  - Ryggsekkproblemet (Knapsack problem).

### Ferdighetsmål

Studenten skal kunne:

- Bruke eksisterende biblioteker for algoritmer og datastrukturer,
- Implementere kjente datastrukturer, inkludert
  - Tabell-lister, lenkede lister,
  - Binære søketrær og heaps,
  - Hash-tabeller,
  - Grafer.
- Implementere kjente algoritmer, inkludert
  - Insertion sort, selection sort,
  - Merge sort, Heap sort.
- Drøfte worst-case ressursbruk for konkrete elementer i dataprogrammer ved hjelp av O-notasjon.
- Bruke grunnleggende generisk programmering i Java.

### Generelle kompetansemål

Studenten skal

- Beherske klassisk asymptotisk analyse av dataprogrammer,
- Kunne bruke eksisterende bibliotek og egenutviklede algoritmer og datastrukturer til å løse praktiske problemer,
- Beherske et språk og begrepsapparat som egner seg for å drøfte ressursbruken til dataprogrammer,
- Ha den kunnskapen om algoritmer og datastrukturer som kreves i videre informatikkstudier.

### Eventuelle forkunnskapskrav

PG1100 og PG2100 eller tilsvarende grunnleggende programmeringskunnskap.

### Pensum

| Tittel     | ISBN              |
|------------|-------------------|
| Algorithms | 978-0-321-57351-3 |

## Undervisningsform

Forelesninger (ca. 50%) og øvinger (ca. 50%)

## Annen informasjon som er relevant.

### Vitnemålstekst

Emnet skal gi innsikt i algoritmer og datastrukturer som er sentrale i arbeidet med implementasjon og design av effektive datasystemer.

## PG4400 C++ Programmering

### Faglig innhold

#### Kort om emnet

Emnet skal gi studentene kunnskap om fundamentale og avanserte programmeringsbegreper i språket C++. Målet er å videreutvikle studentenes programmeringskunnskaper til det som er nødvendig for å utvikle effektive og komplekse systemer, inklusive spill og interaktive applikasjoner. De vil lære å beherske Microsoft Visual Studio som IDE for C++.

#### Kunnskap

- Kjenne til referanser og pekere, samt deres relasjon til hverandre.
- Kjenne til et bibliotek for 2D grafikkprogrammering.
- Kjenne til grunnleggende konsepter i 2D grafikkprogrammering.
- Kjenne til minneallokering i C++ inklusiv forskjellene på stack og heap.
- Kjenne til erstatninger for function pointers i C++, som callback functions og functors.
- Kjenne til templates og deres bruksområder.
- Kjenne til ulike revisjoner av C++ og de viktigste forskjellene.
- Kjenne til hvordan kompilatoren utnytter CPU, minne og cache.

#### Ferdigheter

- Beherske Visual Studio som C++ programmeringsmiljø (opprette projects, kompilere, linke, kjøre).
- Beherske debugging i Visual Studio.
- Programmere med standard IO (cin, cout).
- Beherske skrivning og lesing til og fra fil med streams.
- Anvende pekere og referanser.
- Korrekt anvendelse av minne ved new og delete.
- Korrekt anvendelse av const og mutable.
- Programmere med containers, algoritmer og iteratorer fra STL.
- Kunne anvende og utvikle egne templates.
- Kunne anvende objektorientert programmering i C++ med bruk av klasser, objekter, arv og polymorfi.
- Programmere med threads og synkroniseringsmekanismer i C++.
- Anvende smart pointers for å forenkle minnehåndtering.
- Programmere med operator-overlasting.
- Håndtere og lage egne exceptions.
- Kunne inkludere og anvende tredjepartsbiblioteker ved hjelp av macroer i Visual Studio.
- Beherske enkel 2D grafikkprogrammering.
- Benytte callback functions.
- Kunne anvende tid som en faktor under kjøring av et program.
- Håndtere brukerinput interaktivt i applikasjoner.
- Utnytte "move semantics" for bedre kode.
- Benytte tråder, promises og futures til paralell programmering.

## Generell kompetanse

- Utvide sin forståelse for hvordan objektorientering og template programming kan benyttes til å skape veldesignede programmer.
- Forstå poenget med operator-overlastning og bruksområde.
- Forstå når man skal bruke tredjeparts biblioteker og når en skal utvikle selv.
- Forstå hvordan parallel programmering er nødvendig for å få optimal ytelse av moderne hardware.

## Eventuelle forkunnskapskrav

Objektorientert programmering som fra PG2100 (Programmering 2), samt C-programmering som fra PG3400 (Programmering i C for Linux). Eventuelt tilsvarende emner.

## Pensum

|                              |                   |
|------------------------------|-------------------|
| ISBN                         | ISBN              |
| The C++ Programming Language | 978-0-321-56384-2 |

## Undervisningsform

Forelesninger og øvinger, til sammen 4 timer/uke. Kurset gjennomføres med 12 forelesninger a ca 2 timer og 12 lærerstyrte øvinger a ca 2 timer. Øvingene er ikke obligatoriske, men for å oppnå kompetansemålene forventes det at studentene fullfører øvingsoppgavene på egenhånd der øvingstimene ikke strekker til.

## Annen informasjon som er relevant.

### Vitnemålstekst

Emnet har gitt studentene kunnskap om fundamentale og avanserte programmeringsbegreper i språket C++. Studentenes programmeringskunnskaper er videreutviklet til det som er nødvendig for å utvikle effektive og komplekse systemer, inklusive spill og interaktive applikasjoner. De har lært å beherske Microsoft Visual Studio som IDE for C++.

## PG4500 Game AI

### Faglig innhold

#### Kort om emnet

Etter fullført emne har studentene fått en innføring i kunstig intelligens for spill: Hvilke byggesteiner som utgjør AI i ulike typer spill, og hvilken teknikk som passer i en bestemt situasjon. Videre har studentene fått praktisk erfaring med å implementere kunstig intelligens.

#### Kunnskap:

- Kjenne til forskjeller og likheter mellom kunstig intelligens (AI) for spill og mer tradisjonelle bruksområder.
- Vite hva pathfinding er, samt kunne sammenlikne en rekke kjente pathfinding algoritmer.
- Kjenne til hierarkisk pathfinding.
- Kjenne til fuzzy logic, og hvordan dette kan brukes i AI sammenheng.
- Vite hva regelbaserte systemer er, og hvordan de virker.
- Kjenne til måter AI kan ta taktiske og strategiske avgjørelser på.
- Vite hvordan vi kan lage koordinert AI for grupper.
- Kjenne til teknikker for å la AI systemer lære, blant annet ved hjelp av nevrale nettverk.



### Ferdigheter:

- Kunne lage og bruke agentbasert AI.
- Kjenne til steering behaviors, og bruke de vanligste steering behaviors i praksis.
- Kunne lage og bruke tilstandsmaskiner, særlig i forbindelse med spill.
- Kunne implementere riktig AI for spill innen flere sjangere.
- Kunne programmere A\* algoritmen med flere for pathfinding.
- Kunne scripte AI-oppførsel for spill.

### Generell Kompetanse:

- Forstå hvordan AI kan brukes til å gi unike spillopplevelser.
- Forstå hvilke AI-teknikker som er riktig å velge, gitt bestemte situasjoner.

### Eventuelle forkunnskapskrav

Erfaring med grunnleggende C# programmering (som C# delen av PG3300 programvarearkitektur, eller liknende).

### Pensum

| Tittel                            | ISBN              |
|-----------------------------------|-------------------|
| Artificial Intelligence for Games | 978-0-12-374731-0 |

### Undervisningsform

Forelesninger, øvinger og egenstudier.

### Annen informasjon som er relevant.

#### Vitnemålstekst

Etter fullført emne har studentene fått en innføring i kunstig intelligens for spill: Hvilke byggesteiner som utgjør AI i ulike typer spill, og hvilken teknikk som passer i en bestemt situasjon. Videre har studentene fått praktisk erfaring med å implementere kunstig intelligens.

## BU5100 Entreprenørskap

### Faglig innhold

#### Kort om emnet

Hensikten med emnet er å gi forståelse for hva innovasjon og entreprenørskap er og kunne utvikle ferdigheter for å fremme ideer og realisere disse. Kurset er orientert mot praktisk anvendelse av teknikk og metoder, med utarbeidelse av forretningsplan som bærende element. Etter fullført emne skal studenten ha tilstrekkelig innsikt og kompetanse til å kunne delta i og drive en slik prosess med utvikling av forretningsplan for å etablere et nytt firma eller forretningsområde.

#### Kunnskaper

Etter å ha fullført emnet skal studenten:

- Kunne definere innovasjon og entreprenørskap
- Kjenne til betydningen av innovasjon og entreprenørskap for individ, organisasjoner og samfunn
- Kjenne til de markedsmessige og organisatoriske konsekvenser for innovasjon og entreprenørskap
- Ha forståelse for immaterielle rettigheter

#### Ferdigheter

Etter å ha fullført emnet skal studenten:

- Kunne utvikle planer for innovasjon og entreprenørskap, fra idé til implementering og drift
- Kunne evaluere, velge og selge inn egne prosjekter for egnet målgruppe

## Generell kompetanse

Etter å ha fullført emnet skal studenten:

- Kunne reflektere over innovasjon og entreprenørskap i et videre perspektiv
- Ha kunnskap om og forståelse for hvordan ideer aksepteres og spres i et sosialt system

## Eventuelle forkunnskapskrav

Ingen

## Pensum

| Tittel  | ISBN          |
|---|---------------|
| Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers                           | 9780470876411 |
| The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses | 9780670921607 |

## Undervisningsform

Forelesninger og øvinger

## Annen informasjon som er relevant.

### Vitnemålstekst

Emnet har gitt studenten forståelse for hva innovasjon og entreprenørskap er, og ferdigheter med å fremme ideer og realisere disse. Kurset er orientert mot praktisk anvendelse av teknikk og metoder, med utarbeidelse av forretningsplan som bærende element.

## PG5200 Tools programmering

### Faglig innhold

#### Kort om emnet

Målet med faget er å gi studentene innsikt i utfordringer og løsninger for utvikling av verktøy som lar andre profesjoner produsere innhold for spill. Studentene utvikler en enkel leveleditor og andre verktøy som trengs for å utvikle spill og andre multimedieprodukter. Videre vil studentene bli kjent med behovet for stabilitet, feilhåndtering og hurtig reaksjon.

#### Kunnskap:

- Forstå grunnleggende funksjonalitet i en spillmotor, og hva konseptet "toolchain" innebærer.
- Kjenne til forskjellige typer spillmotorer samt forskjellen mellom rene grafikkmotorer og komplette spillmotorer.
- Forstå hvorfor gode verktøy er essensielle for effektiv produksjon av innhold.
- Forklare de forskjellige kategoriene av verktøy brukt i spillutvikling inkludert leveleditors, propertyeditor og support tools.
- Beskrive den viktigste funksjonaliteten i en leveleditor.
- Kjenne til tredjeparts-løsninger for spillverktøy.
- Kjenne til nettverksprotokoller og hvordan spill kommuniserer over nett.

#### Ferdigheter:

- Vurdere nytten av å lage utviklingsverktøy selv i forhold til å benytte tredjeparts-løsninger.
- Utvikle kravspesifikasjon for et variert utvalg verktøy. Programmere verktøy i C# med WPF.
- Implementere essensiell funksjonalitet for en leveleditor.

- Bruk og håndtering av exceptions for bedret stabilitet.
- Unngå tunge operasjoner i GUI-tråd.
- Kunne legge inn support for scripting i eksisterende kode.
- Kunne lagre spilltilstand og logge hendelser i en database.
- Kunne serialisere og deserialisere spilldata.

#### Generell kompetanse:

- Forstå hvordan en god verktøykjede kan forbedre utviklingseffektiviteten.
- Reflektere kritisk rundt spillutviklingsprosessen og de nødvendige krav.

#### Eventuelle forkunnskapskrav

Erfaring med grunnleggende C# programmering (gjennom C# elementene i PG3300 programvarearkitektur, eller tilsvarende)

#### Pensum

| Tittel                            | ISBN              |
|-----------------------------------|-------------------|
| Professional C# 2012 and .NET 4.5 | 978-1-118-31442-5 |

#### Undervisningsform

12 forelesninger med øvinger

#### Annen informasjon som er relevant.

##### Vitnemålstekst

Studenten har fått innsikt i utfordringer å løsninger for utvikling av verktøy som lar andre profesjoner produsere innhold for spill. Studenten har utviklet en enkel leveleditor og andre verktøy som trengs for å utvikle spill og andre multimedieprodukter. Videre er studenten kjent med behovet for stabilitet, feilhåndtering og hurtig reaksjon.

## RF5100 Lineær algebra

### Faglig innhold

#### Kort om emnet

Datamodellering av geometriske objekter avhenger i stor grad av vektorgeometri og lineær algebra. I dette kurset skal studenten forberede seg til å arbeide med tredimensjonal grafikkprogrammering ved få nødvendig innsikt i geometri, transformasjonsteori og lineær algebra. Ved siden av dette, er det et viktig mål at studenten skal få en generell trening i matematiske metoder.

#### Kunnskapsmål:

Studenten skal få kunnskap om

- Representasjon av geometriske objekter ved hjelp av vektorer,
- Lineære transformasjoner,
- Affine transformasjoner,
- Geometriske transformasjoner som translasjon og rotasjon,
- Komplekse tall, kvaternioner og Euler-vinkler
- Parallellprojeksjon og sentralprojeksjon,
- Lineære ligningssystemer,
- Indreproduktrom og approksimasjoner,
- Ulike anvendelsesområder for lineær algebra.

## Ferdighetsmål:

Studenten kunne

- Arbeide med vektorer, matriser og lineære ligningsystem,
- Kombinere og veksle mellom ulike koordinatsystemer,
- Bruke lineær algebra i programmering,
- Bruke lineær algebra på ulike anvendelsesområder.

## Generelle kompetansemål:

- Studenten skal ha god oversikt over vektorgeometri og grunnleggende lineær algebra med anvendelser.
- Studenten skal ha et godt utgangspunkt for å arbeide videre med grafikkprogrammering.
- Studenten skal utvikle sin matematiske modenhet.

## Eventuelle forkunnskapskrav

PG1100 og PG2100 eller tilsvarende grunnleggende programmeringskunnskap.

RF3100 Matematikk og fysikk

## Pensum

| Tittel  | ISBN          |
|---|---------------|
| 3D Math Primer for Graphics and Game Development, 2nd Edition, 2011 | 9781568817231 |

## Undervisningsform

Foreslesninger (ca. 50%) og øvinger (ca. 50%)

## Annen informasjon som er relevant.

### Vitnemålstekst

Emnet gir innsikt i vektorgeometri og grunnleggende lineær algebra med anvendelser.

## PG6200 Grafikkprogrammering

### Faglig innhold

#### Kort om emnet

Hensikten med dette emne er å gi en god oversikt over hvordan grafikk programmeres med C++ og moderne OpenGL, samt de sentrale teknikker og metoder som brukes. Emnet skal videre gi innsikt i avanserte metoder for grafikk og rendering (multipass-algoritmer, avansert lyssetting og skygger, mm.), blant annet ved bruk av nettopp OpenGL og OpenGL Shading Language. Videre lærer man om effektiv represe

### Kunnskapsmål:

Studenten skal få kunnskap om blant annet:

- Representasjon av geometri
  - Farger, lysegenskaper og materialelegenskaper
- Buffer objects
  - Framebuffer objects
- Multipass-algoritmer
- Per-pixel lyssetting
- Blending
- Teksturering
- Skygger
- Cube maps

- Normal-mapping
- Subdivisjon.

#### Ferdighetsmål:

Studenten skal blant annet:

- Arbeide med geometriske transformasjoner,
- Benytte kvaternioner, vektorer og matriser ifm. grafikkprogrammering,
- Arbeide med ulike projeksjoner ifm. grafikkprogrammering.

#### Generelle kompetansemål:

- Få en god oversikt over OpenGL og OpenGL "state machine" (tilstandsmaskin).
- Kunnskap om 3D-grafikkprogrammering, bruk av shadere og avanserte emner innen grafikk og rendering.
- Kjennskap til hovedtrekk i utviklingen av OpenGL, samt utviklingen av grafikkmaskinvare og moderne grafikkort-arkitektur. • Via lab-øvelser bli komfortabel med å programmere OpenGL i Visual Studio med C++.
- Få praktisk erfaring med hvordan OpenGL fungerer gjennom modifisering og utvidelse av utdelt «skjelettkode».ntasjon og behandling av geometri.

#### Eventuelle forkunnskapskrav

PG1100 og PG2100 eller tilsvarende grunnleggende programmeringskunnskap.

PG4400 C++ Programmering

RF5100 Lineær algebra

RF3100 Matematikk og fysikk

Eller tilsvarende

#### Pensum

| Tittel  | ISBN       |
|---|------------|
| Real-Time Rendering, Third Edition                              | 1568814240 |
| OpenGL Programming Guide: The Official Guide to Learning OpenGL | 0321773039 |

#### Undervisningsform

Forelesning og øvinger.

#### Annen informasjon som er relevant.

##### Vitnemålstekst

Emnet gir kunnskap innen avanserte metoder for 3D-grafikk, hvor hovedfokus ligger på moderne shader-programmering og effektiv rendering. Praktisk grafikkprogrammering skjer med C++ og OpenGL

#### PJ6000 Hovedprosjekt

##### Faglig innhold

##### Kort om emnet

Studenten skal få yrkeserfaring ved å gjennomføre et prosjekt i en bedrift, etablere eget selskap eller delta i forskningsprosjekt. Studenten skal demonstrere bred kunnskap om sentrale temaer og teorier, og vise ferdigheter i metoder, verktøy og teknologi innenfor fagområdet.

## Kunnskaper

Etter å ha fullført emnet skal studenten:

- Kunne forklare hvordan oppdraget er til nytte for virksomheten/bedriften/oppdragsgiver
- Kunne planlegge og styre gjennomføringen av prosjektet
- Kunne anvende kilder på en korrekt måte

## Ferdigheter

Etter å ha fullført emnet skal studenten:

- Kunne innhente nødvendig informasjon på en strukturert måte
- Kunne planlegge og gjennomføre et prosjekt i en bedrift etter valgt metodikk
- Kunne utforme og kvalitetssikre løsningen i henhold til spesifikasjoner og etablerte standarder
- Kunne formidle sentralt fagstoff, teorier og løsninger både skriftlig, muntlig og gjennom andre relevante uttrykksformer

## Generell kompetanse

Etter å ha fullført emnet skal studenten:

- Kunne beskrive og vurdere erfaringer i prosjektet, og hva som er lært av prosessen.
- Kunne formidle resultatet av prosjektet på en klar og velstrukturert måte, tilpasset målgruppen.
- Kunne vurdere hvordan oppdraget forholder seg til etablert kunnskap/forskning på området

## Eventuelle forkunnskapskrav

Det kreves at alle emner fra 1.klasse er bestått, samt bestått prosjektemne i software engineering.

## Pensum

| Tittel                                     | ISBN          |
|--|---------------|
| Project Management for Information Systems | 9780132068581 |
| Prosjektoppgaven - krav til utforming      | 9788215003641 |

## Undervisningsform

Se egne retningslinjer for hovedprosjekt på emnesiden

## Annen informasjon som er relevant.

### Vitnemålstekst

Bachelorstudiet avsluttes med hovedprosjektet som innebærer å gjennomføre et IT-prosjekt i en bedrift. Prosjektet gir yrkeserfaring hvor kunnskap og ferdigheter fra hele studiet skal anvendes.

## PJ6100 Undersøkellesmetoder

### Faglig innhold

#### Kort om emnet

Emnet skal gjøre studenten kjent med metoder med særlig vekt på metoder som har relevans for IT-bransjen. Emnet er et støtteemne til hovedprosjektet, primært i forhold til rapportskrivning og metode.

#### Kunnskaper:

Etter å ha fullført emnet skal studenten kunne:

- Gjøre rede for forskjellen mellom kvalitative og kvantitative metoder
- Gjøre rede for grunnprinsippene i metoder som kvalitative intervjuer, fokusgruppeundersøkelser, brukertesting, surveyundersøkelser og Web-/ data mining.

## Ferdigheter:

Etter å ha fullført emnet skal studenten kunne:

- Gjennomføre en undersøkelse som er relevant for et IT-prosjekt
- Begrunne valg av metode ut fra en gitt faglig problemstilling
- Skrive en metoderapport som kan inngå som del av PJ6000 hovedprosjekt
- Gjennomføre litteratursøk og utarbeide en korrekt referanseliste

## Generell kompetanse:

- Vurdere og benytte relevant metodikk for et IT-prosjekt.

## Eventuelle forkunnskapskrav

Ingen

## Pensum

| Tittel                    | ISBN          |
|---------------------------|---------------|
| Metode og oppgaveskriving | 9788205423985 |

## Undervisningsform

Emnet gjennomføres som forelesninger og workshops.

## Annen informasjon som er relevant.

### Vitnemålstekst

Emnet gir en innføring i grunnleggende metoder som er relevante for IT-bransjen.

## PG5500 Embedded systems

### Faglig innhold

#### Kort om emnet

Hensikten med dette emnet er å gi kunnskap i og om bruken av embedded systems med tilhørende arkitektur. Emnet skal også gi kunnskap om grunnleggende hardwarekomponenter og Arduino. Emnet skal gi studentene ferdigheter i å programmere applikasjoner som kommuniserer med IO enheter tilknyttet mikrokontrollere. Studenten skal ha kompetanse i bruksområder og praktiske anvendelser for embedded systems.

#### Kunnskap:

- Forstå uttrykket Embedded systems og kjenne til forskjellen mellom en CPU og en mikrokontroller
- Kjenne til de mest brukte arkitekturene for embedded systems
- Ha kunnskap om hardwaren i Raspberry Pi, og hvordan den brukes
- Ha kunnskap Arduino og hvordan den brukes

#### Ferdigheter:

- Kunne sette opp og installere OS og annen software på Raspberry PI
- Kjenne til og bruke grunnleggende elektroniske komponenter som LED, knapper, motstand, Buzzer og transistorer.
- Kunne styre eksterne komponenter, med både analoge og digitale signaler, fra embedded kontroller (segment display, led matrise, lcd, i/o expansion)
- Kunne lese inn data fra analoge og digitale eksterne sensorer. (eks. lys, temperatur, knapper)
- Kunne styre motorer (servo, DC, stepper, solenoid m.m.)
- Kommunisere trådløst med IR

- Bruke data fra eksterne sensorer til å styre eksterne komponenter
- Kjenne til og bruke viktige protokoller i embedded systems som bit-banging, SPI, I2C, 1Wire og UART.
- Forstå hvordan embedded systemer virker i grensesnittet mellom elektronikk og program (Eeprom, interrupts, minnehåndtering)
- Kunne designe enkle krets-kort med Fritzing

#### Generell kompetanse:

- Ha innsikt i bruksområder og praktiske anvendelser for embedded systems
- Designe, teste og kritisk evaluere Embedded Systems som løsningene på virkelige problemstillinger som å lage roboter og spillkonsoller

#### Eventuelle forkunnskapskrav

Gode ferdigheter i programmering (min tilsvarende 2.kl Bachelor)

#### Undervisningsform

Forelesninger, øvinger og egenstudium

#### Annen informasjon som er relevant.

##### Vitnemålstekst

Hensikten med dette emnet er å gi kunnskap i og om bruken av embedded systems med tilhørende arkitektur. Emnet skal også gi kunnskap om grunnleggende hardwarekomponenter og Arduino. Emnet skal gi studentene ferdigheter i å programmere applikasjoner som kommuniserer med IO enheter tilknyttet mikrokontrollere. Studenten skal ha kompetanse i bruksområder og praktiske anvendelser for embedded systems.

## PG2201 Unity Utvikling

### Faglig innhold

#### Kort om emnet

Kurset omhandler programmering og utvikling med Unity spillutviklingsplattformen. Studentene vil bli kjent med utviklingsmiljøets muligheter og effektiv bruk av de ulike komponentene. Studentene skal bli i stand til å importere, bruke og scripte forskjellige spill-objekter ("assets") inn i en "scene" for dermed å opprette "levels" med grunnleggende spill-adferd.

#### Kunnskap:

- Forstå uttrykket Embedded systems og kjenne til forskjellen mellom en CPU og en mikrokontroller
- Kjenne til de mest brukte akritekturene for embedded systems
- Ha kunnskap om hardwaren i Raspberry Pi, og hvordan den brukes
- Ha kunnskap Arduino og hvordan den brukes

#### Ferdigheter:

- Kunne sette opp og installere OS og annen software på Raspberry PI
- Kjenne til og bruke grunnleggende elektroniske komponenter som LED, knapper, motstand, buzzer og transistorer.
- Kunne styre eksterne komponenter, med både analoge og digitale signaler, fra embedded kontroller (segment display, led matrise, lcd, i/o expansion)
- Kunne lese inn data fra analoge og digitale eksterne sensorer. (eks. lys, temperatur, knapper)
- Kunne styre motorer (servo, DC, stepper, solenoid m.m.)
- Kommunisere trådløst med IR
- Bruke data fra eksterne sensorer til å styre eksterne komponenter



- Kjenne til og bruke viktige protokoller i embedded systems som bit-banging, SPI, I2C, 1Wire og UART.
- Forstå hvordan embedded systemer virker i grensesnittet mellom elektronikk og program (Eeprom, interrupts, minnehåndtering)
- Kunne designe enkle kretsdiagrammer med Fritzing

#### Generell kompetanse:

- Ha innsikt i bruksområder og praktiske anvendelser for embedded systems
- Designe, teste og kritisk evaluere Embedded Systems som løsninger på virkelige problemstillinger som å lage roboter og spillkonsoller

#### Eventuelle forkunnskapskrav

PG1100

#### Pensum

| Tittel                                | ISBN       |
|---------------------------------------|------------|
| Unity 3.x Game Development Essentials | 1849691444 |

#### Undervisningsform

Forelesninger, øvinger, prosjekter og hjemme-eksamen.

#### Annen informasjon som er relevant.

#### Vitnemålstekst

Emnet gir en introduksjon til spill-utvikling og -programmering med Unity spillmotoren.