

## **Studieplan**

**Start IT - introduksjon til programmering og IT-utvikling**

**30 studiepoeng**

Tekst.....	Sidetall
1 Navn på fagskoleutdanningen .....	3
2 Innledning .....	3
3 Opptakskrav .....	4
4 Omfang og forventet arbeidsmengde.....	6
5 Organisering og oppbygging av studiet.....	6
6 Overordnet læringsutbytte .....	7
7 Læringsutbyttebeskrivelser for hvert emne .....	8
8 Beskrivelse av sammenhengen mellom læringsutbyttebeskrivelsene på emnenivå og det totale læringsutbytte for utdanningen.....	17
9 Undervisningsformer og læringsaktiviteter.....	23
Beskrivelse av undervisningsformene og læringsaktivitetene .....	23
Pedagogisk opplegg .....	26
Veiledning og oppfølging.....	27
10 Arbeidskraemneeksamen og vurdering .....	29
11 Litteraturliste/læremidler .....	30
Vedlegg 1 Undervisningsplan.....	31

## 1 Navn

Start IT - introduksjon til programmering og IT-utvikling.

## 2 Innledning

Start IT - introduksjon til programmering og IT-utvikling er en fagskoletutdanning som gjennomføres over et halvt år på heltid på nivå 5.1 i NKR med 30 studiepoeng. Med denne utdanningen, vil vi tilby studentene en praktisk og teoretisk tilnærming til IT-faget med de viktigste prinsippene innen strukturert og objektorientert programmering og samtidig hva IT-utvikling er utover ren programmering. Temaene som går gjennom er valgt ut på bakgrunn av det typiske behovet for IT-utvikling i norsk næringsliv som er meldt fra våre samarbeidspartnere.

### **Bakgrunn:**

«Norges behov for IKTkompetanse i dag og framover Rapport 1-2021» er utarbeidet på oppdrag for Abelia, IKT-Norge, NITO, Digital Norway, Tekna, Negotia og EL og IT Forbundet. Organisasjonene er alle opptatt av at Norge utvikler nok IKT-kompetanse til å møte framtidige behov. På denne bakgrunn ble Samfunnsøkonomisk analyse AS bedt om å analysere etterspørselen etter og tilbudet av IKT-kompetanse i det norske arbeidslivet, i dag og framover. De siste to tiårene har verden opplevd en teknologisk revolusjon knyttet til IKT, med digitalisering av stadig større deler av informasjonsflyten i samfunnet. Verden er fremdeles i en pågående endring knyttet til denne teknologiske revolusjonen. Med bruk og implementering av IKT følger endringer i både hvilke oppgaver arbeidskraften har, og hvordan disse løses. Dette øker behovet for IKT-kompetanse, knyttet til alt fra utvikling, til produksjon, installasjon og vedlikehold av både maskin- og programvare. Denne rapporten redegjør for SØAs analyse av etterspørselen etter og tilbudet av IKT-kompetanse i det norske arbeidslivet. Veksten i bruk av IKT øker behovet for IKT-kompetanse i arbeidsmarkedet. Digitaliseringen de siste 30 årene har ført til betydelig vekst i både i næringer som produserer IKT eller som bidrar til at IKT blir tatt i bruk, i antall sysselsatte med formell IKT-utdanning og i antall lønnskakere med IKT-yrker. IKT-næringene er blant de raskest voksende næringene i landet.

I Omstillingsbarometeret som Abelia har gitt ut 16.08.2022 opplyses det at innen teknologi- og digitalisering har Norge mistet sin ledende posisjon og faller tilbake på nær alle dimensjoner knyttet til teknologi og digitalisering. Andre land digitaliserer i et hurtigere tempo enn oss, både i privat og offentlig sektor.

I IT-utvikleryrket møter man mange komplekse problemstillinger, og det er avgjørende å kunne stå i det vanskelige. Vår pedagogiske metode er å jobbe mye med å styrke studentenes kompetanse på seg selv i forhold til læring og tenkning, samarbeid og kommunikasjon og refleksjon og selvledelse.

### **Ansvars- og funksjonsområde:**

Utdanningen tar sikte på å utdanne reflekterte yrkesutøvere i programvareutvikling med gode kunnskaper og ferdigheter om programmering og IT-utvikling. Studentene får også kunnskap om viktige temaer innen læring, tenkning, samarbeid, kommunikasjon, selvledelse og refleksjon.

### **Tjenesteområde:**

Som ferdig yrkesutøver med fagskoletutdanning i Start IT - introduksjon til programmering og IT-utvikling, skal kandidaten kunne arbeide som IT-utvikler. Kandidaten skal etter endt utdanning kunne samarbeide med kunder, samt personell innen design, spesifisering, testing, arkitektur og ha evne til tilpasning i faglige virksomheter og arbeidsforhold i stadig endring.

## 3 Opptakskrav

Grunnlaget for opptak skal være relevante kvalifikasjoner på nivå 4 i Nasjonalt kvalifikasjonsrammeverk for livslang læring. Beskriv både hvilke utdanningsprogram i videregående opplæring som utgjør det formelle opptakskravet, og kort hvilke krav som stilles til søkere som tas opp på grunnlag av realkompetanse.

## 1. Formell kompetanse

### Formelle opptakskrav:

Formelt opptakskrav er fullført og bestått videregående opplæring med fagbrev enten i IKT-servicefaget eller IT-utviklerfaget eller Dataelektroniker eller Medieteknikker eller Studiespesialisering i medier og kommunikasjon.

### 2. Opptak på bakgrunn av realkompetanse:

Søkere som ikke tilfredsstillt opptakskravet for formell kompetanse, kan søke om opptak på grunnlag av realkompetanse. Minimumsalder for opptak på bakgrunn av realkompetanse er at søker fyller 23 år i løpet av året det søkes opptak. Realkompetansen vurderes opp mot læringsutbytte på nivå 4 i NKR.

Søkere:

- må levere dokumentasjon på realkompetanse som tilsvarer fullført og bestått videregåendeopplæring i oppgitte fagbrev eller vitnemål fra utdanningsprogram som gjelder for formelt opptakskrav
- må levere dokumentasjon for fellesfag som tilsvarer læreplanene i vg1 og vg2 i yrkesfaglige programområder.

Dokumentasjon:

- må være utformet slik at den kan vurderes mot læringsutbyttebeskrivelsene / kompetansemålene i oppgitt fagbrev eller vitnemål fra utdanningsprogram som gjelder for formelt opptakskrav.

Realkompetansen kan være oppnådd både gjennom formell, ikke-formell eller uformell læring. Dvs. alle kunnskaper og ferdigheter en person har tilegnet seg gjennom utdanning, betalt eller ubetalt arbeid, etterutdanning, fritidsaktiviteter, organisasjonsarbeid og annet som kommer i tillegg til den kompetansen som er dokumentert gjennom grunnutdanningen. Yrkespraksis må være relevant til studiet, og søkeren må ha nødvendige faglige forutsetninger for å gjennomføre studiet.

Utfyllende regler om språkkrav for søkere med utenlandsk utdanning er beskrevet i vår forskrift § 2-4.

### 3. Utforming av realkompetansesøknad:

Søknadsskjema på vår hjemmeside benyttes og dokumentasjon legges ved.

#### Innlevert dokumentasjon må oppfylle minimum ett av de følgende kravene

1. relevant yrkespraksis fra det fagområdet fagskolen omfatter i form av arbeidsattest eller arbeidsbekreftelse som angir start- og evt. sluttidspunkt for arbeidsforholdet, stillingsprosent og /eller arbeidstimer i sum eller pr. uke/måned
2. utdanning og kurs i form av vitnemål, kursbevis, kompetansebevis
3. relevant erfaring fra blant annet organisasjonsliv, verv, fritid- og fritidsinteresser. Dokumentasjon legges ved dersom dokumentbare forhold.
4. Dersom søkeren mangler dokumentasjon på deler av sin kompetanse, kan fagskolen legge til rette for å synliggjøre eller demonstrere den, slik at den dokumenteres, enten av søkeren selv, eller av en som observerer.

Praksis, utdanning og andre forhold som skal gi grunnlag for opptak, må dokumenteres med attesterte kopier. Attester for praksis må angi lengde på arbeidsforhold, stillingsprosent og arbeidsinnhold som er detaljert beskrevet og som kan vurderes mot kompetansemålene i utdanningsprogrammet for formell kompetanse, nivå 4 i Nasjonalt Kvalifikasjonsrammeverk.

Attester må videre være datert for å komme i betraktning. Attester regnes bare fram til datoen de er skrevet ut, selv om søkeren selv opplyser at arbeidsforholdet fortsatte utover dette tidspunkt. Realkompetanse vurderes i forhold til det formelle opptakskravet.

#### **4. Betinget opptak.**

Søkere som ikke har dokumentert fullført og bestått fag- eller svenneprøve innen fristen for å sende inn dokumentasjon, og som derfor ikke er kvalifisert for opptak, kan få opptak til fagskoleutdanning dersom de kan dokumentere at de skal gjennomføre fag- eller svenneprøven i løpet av det påfølgende semesteret. Studenter med opptak på visse vilkår som ikke oppfyller opptakskravene i løpet av første semester etter opptak, mister studieplassen og studieretten.

## 4 Omfang og forventet arbeidsmengde

Utdanningen som er et nettbasert fulltidsstudium, har et omfang på 30 studiepoeng og en samlet normert studietid på et semester. Oppstart for et høstsemester er i uke 33 og avsluttes i uke 51. Oppstart for et vårsemester er i uke 3 og avsluttes i uke 23. Totalt antall normerte arbeidstimer for studentene omfatter all lærerstyrt nettundervisning i digitalt klasserom, veiledning og beregnet egenarbeid for studentene. Det er totalt 800 timer som er fordelt på 20 uker.

SP= Studie- poeng	Emner	Antall uker	Timer med faglærer	Timer egen- arbeid	Timer totalt
10 SP	Emne 1 Grunnleggende programmering i JavaScript på en nettside	7	123	157	280
10 SP	Emne 2 IT-utvikling, prosesser og metodikk	7	116	164	280
10 SP	Emne 3 C# og objektorientert programmering	6	102	138	240
30 SP	Sum	20	341	459	800

## 5 Oppbygning/organisering av utdanningen

Studiet er organisert som et nettbasert fulltidsstudium, som er delt inn i tre enkeltstående og påfølgende emner.

Emne 1 Grunnleggende programmering i JavaScript på en nettside går over 7 uker  
 Emne 2 IT-utvikling, prosesser og metodikk går over 7 uker  
 Emne 3 C# og objektorientert programmering går over 6 uker

Emne 1 Grunnleggende programmering i JavaScript på en nettside introduserer grunnleggende strukturert programmering. Studentene lærer å bygge opp en nettside ved hjelp av HTML og CSS - og endre nettsiden dynamisk ved hjelp av JavaScript.

Emne 2 IT-utvikling, prosesser og metodikk bygger videre på emne 1 Grunnleggende programmering i JavaScript på en nettside. Det studentene har lært om programmering i emne 1 Grunnleggende programmering i JavaScript på en nettside vil i emne 2 IT-utvikling, prosesser og metodikk settes i sammenheng med fokus på IT-utvikling.

Emne 3 C# og objektorientert programmering går videre i programmeringsfaget og introduserer objektorientert programmering og programmeringsspråket C#. Emne 3 C# og objektorientert programmering bygger videre på både emne 1 Grunnleggende

programmering i JavaScript på en nettside og emne 2 IT-utvikling, prosesser og metodikk.

Oppbyggingen vil bidra til at studentene oppnår læringsutbyttet.

## **6 Overordnet læringsutbytte**

### **Kunnskaper:**

#### **Kandidaten**

- har kunnskap om strukturert og objektorientert programmering, utviklingsmetodikk og -prosess, samt ulike verktøy som anvendes innen utvikling av programvare
- har kunnskap om etikk, personvern og GDPR som er relevant for IT-utvikling og programmering
- har innsikt i relevant regelverk i forhold til kvalitetskrav og normer for programmering og IT-utvikling
- har kunnskap om programmering og IT-utvikling og har kjennskap til utvikling av programvare for ulike yrkesfelt
- kan oppdatere sin kunnskap om IT-utvikling gjennom informasjonsinnhenting om trender og teknologier og kontakt med fagmiljøet
- forstår betydningen av programmering og IT-utvikling i et samfunns- og verdiskapingsperspektiv

### **Ferdigheter:**

#### **Kandidaten**

- kan anvende programmering, metodikk, prosess og verktøy til utvikling av programvare
- kan anvende kodeutviklingsmiljø og versjonsstyringssystem i utvikling av programvare
- kan finne informasjon og fagstoff om programmering og IT-utvikling som er relevant for en programvareløsning
- kan kartlegge status for eget arbeid innen IT-utvikling og programmering og identifisere faglige problemstillinger og behov for endring i bruk av kildekode og teknologi

### **Generell kompetanse:**

#### **Kandidaten**

- har forståelse for etiske prinsipper i IT-utviklingsbransjen
- har utviklet en etisk grunnholdning i utøvelsen av programmering og IT-utvikling
- kan utføre utvikling av programvare etter utvalgte sektorers behov
- kan bygge relasjoner med faggrupper innen IT-utvikling og på tvers av sektorer, samt med eksterne utviklere, designere og andre IT-medarbeidere
- kan bidra til IT-utvikling av helhetlige programvareløsninger

## 7 Læringsutbyttebeskrivelse og faglig innhold for hvert emne

### Emne 1 Grunnleggende programmering i JavaScript på en nettside

#### Kunnskaper:

##### Kandidaten

- har kunnskap om HTML, CSS og strukturert programmering, samt kodeutviklingsmiljø og versjonsstyringssystem som anvendes innen grunnleggende programmering i JavaScript på en nettside
- har kunnskap om design mønsteret model-view-controller og enhetstesting som anvendes innen utvikling av nettbasert programvare
- har kunnskap om konkrete problemløsningsteknikker som anvendes innen IT-utvikling og programmering
- har kunnskap om refactoring som anvendes innen strukturert programmering
- har kunnskap om etikk, personvern og GDPR som er relevant for programmering
- har innsikt i relevant regelverk i forhold til kvalitetskrav og normer for grunnleggende programmering i JavaScript på en nettside
- kan oppdatere sin kunnskap om grunnleggende programmering i JavaScript på en nettside gjennom informasjonsinnhenting om trender og teknologier og kontakt med fagmiljøet

#### Ferdigheter:

##### Kandidaten

- kan anvende koding i HTML og CSS og strukturert programmering til utvikling av grunnleggende programmering i JavaScript på en nettside
- kan anvende kunnskap om refactorings-teknikker i strukturert programmering
- kan anvende design mønsteret model-view-controller i egen programmering i JavaScript på en nettside
- kan anvende kodeutviklingsmiljø og versjonsstyringssystem i utvikling og feilsøking av programvare
- kan anvende problemløsningsteknikker til utvikling og feilretting av programvare
- kan anvende enhetstesting til kontroll og kvalitetssikring av programvare
- kan finne informasjon og fagstoff som er relevant for programmering i JavaScript på en nettside
- kan kartlegge status for eget arbeid innen programmering i JavaScript på en nettside og identifisere faglige problemstillinger og behov for endring i bruk av kildekode og teknologi

#### Generell kompetanse:

##### Kandidaten

- har forståelse for etiske prinsipper i grunnleggende programmering i JavaScript på en nettside
- har utviklet en etisk grunnholdning i utøvelsen av grunnleggende programmering i JavaScript på en nettside
- kan utføre grunnleggende programmering i JavaScript på en nettside ut fra behovet i et oppdrag



- kan bygge relasjoner med faggrupper innen grunnleggende programmering i JavaScript på en nettside og på tvers av sektorer, samt med eksterne utviklere, designere og andre IT-medarbeidere
- kan bidra med grunnleggende programmering i JavaScript på en nettside som en del av helhetlige programvareløsninger

#### **Tema:**

- HTML
- CSS
- Versjonsstyring ved hjelp av GIT og GitHub.com

#### **Programmering i JavaScript**

- Document Object Model
- Variabler og operatører
- Funksjoner, parametre og returverdi
- Design patternet model-view-controller
- Valgsetninger og logiske verdier
- Enhetstesting
- Problemløsningsteknikker
- Løkker
- Lister
- Objekter

#### **Undervisningsformer og læringsaktiviteter:**

- Nettundervisning i digitalt klasserom
- Gruppearbeid i digitalt klasserom
- Veiledning en-til-en og i gruppe
- Felles oppstart og morgenmøte
- Refleksjonslogg med skriftlige tilbakemeldinger
- Egenarbeid

#### **Veiledning og oppfølging:**

Det gis totalt 84 timer veiledning til studentene. Dette er fordelt med 12 timer veiledning pr. uke i 7 uker. Veiledningen er på oppgaver og arbeidskrav og gjennomføres etter avtale med studentene og fordeles på veiledning individuelt og i grupper der det er hensiktsmessig.

#### **Arbeidskrav og avsluttende emneeksamen for emne 1 Grunnleggende programmering i JavaScript på en nettside**

1 obligatorisk oppgave skal løses og leveres inn i form av kildekode og et skriftlig dokument. Studenten skal drøfte og reflektere i lys av relevant faglitteratur. Oppgavens omfang: 600 ord (10% +/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstr. 12, Times New Roman og 1.5 linjeavstand.

#### **Vurderingsform for arbeidskrav og avsluttende emneeksamen for emne 1 Grunnleggende programmering i JavaScript på en nettside**

Arbeidskrav og avsluttende emneeksamen i form av en skriftlig innlevering vurderes med bestått/ikke bestått. Studenten får en skriftlig tilbakemelding samt bestått/ikke

bestått innen 3 uker etter at innlevering ble mottatt. Karakteren bestått er emnekarakter som føres på vitnemålet.

### **Vurderingskriterier for arbeidskrav og avsluttende emneeksamen for emne 1 Grunnleggende programmering i JavaScript på en nettside**

Vurderingskriterier for arbeidskrav og avsluttende emneeksamen gis i forhold til 2-4 valgte læringsutbyttebeskrivelser i emne 1 Grunnleggende programmering i JavaScript på en nettside. Formelle krav: Oppgavens omfang: 600 ord (+/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstørrelse 12, Times New Roman og linjestørrelse 1.5. Besvarelsen leveres elektronisk innen fastsatt tid anonymt for sensor.

#### **Litteratur:**

ISBN 9788202661243 Koding for alle i JavaScript, Terje Kolderup, Cappelen Damm Akademisk, 2020

Kapittel 1 - 13, 176 sider

ISBN 978-3319051543 Agile!: The Good, the Hype and the Ugly, Bertrand Meyer, Springer, 2014

Kapittel 1 - 9, 144 sider

ISBN 9781119795780 Beginning C# and .NET, 2021 Edition, Benjamin Perkins, Jon D. Reid, Wiley, 2021

Kapittel 1 - 12, 354 sider

## Emne 2 IT-utvikling, prosess og metodikk

### Kunnskaper:

#### Kandidaten

- har kunnskap om systemutviklingsmetodikk som anvendes innen IT-utvikling
- har kunnskap om utviklingsprosessen som anvendes innen IT-utvikling
- har kunnskap om verktøy for prototypiering av skjermbilder som anvendes innen IT-utvikling
- har kunnskap om model-view-controller og objekter som anvendes ved datamodellering
- har kunnskap om etikk, personvern og GDPR som er relevant for IT-utvikling
- har innsikt i relevant regelverk i forhold til kvalitetskrav og normer for programmering og IT-utvikling
- har kunnskap om programmering og IT-utvikling og har kjennskap til utvikling av programvare for ulike yrkesfelt
- kan oppdatere sin kunnskap om IT-utvikling gjennom informasjonsinnhenting om trender, teknologier, prosess og metodikk og kontakt med fagmiljøet
- forstår betydningen av programmering og IT-utvikling i et samfunns- og verdiskapingsperspektiv

### Ferdigheter:

#### Kandidaten

- kan anvende kunnskap om systemutviklingsmetodikk til utvikling av programvare
- kan anvende kunnskap om utviklingsprosessen i utvikling av programvare
- kan anvende kunnskap om model-view-controller og objekter i datamodellering
- kan anvende kunnskap om etikk, personvern og GDPR som er relevant for IT-utvikling
- kan anvende metodikk, prosess og verktøy til utvikling av programvare
- kan anvende verktøy for prototypiering av skjermbilder i IT-utvikling
- kan finne informasjon om trender, teknologier, prosess og metodikk som er relevant for en programvareløsning
- kan kartlegge et oppdrag fra en kunde og identifisere faglige problemstillinger og behov for prosess, metodikk, programmering og IT-utvikling

### Generell kompetanse:

#### Kandidaten

- har forståelse for etiske prinsipper i IT-utviklingsbransjen
- har utviklet en etisk grunnholdning i utøvelsen av programmering og IT-utvikling
- kan utføre prosess, metodikk, programmering og IT-utvikling etter kundens behov
- kan bygge relasjoner med fagfeller og på tvers av fag, samt med kunder og brukere av programvaren
- kan bidra med prosess, metodikk, programmering og IT-utvikling som en del av helhetlige programvareløsninger

**Tema:**

- Utviklingsprosessen
- Kundedialog
- Ulike systemutviklingsmetodikker gjennom historien
- Smidige metoder, Scrum
- Analyse og design
- Tegning av skjermbilder
- Model-view-controller
- Objekter, abstrakte datastrukturer og modellering
- Etikk
- Personvern – GDPR

**Undervisningsformer og læringsaktiviteter:**

- Nettundervisning i digitalt klasserom
- Gruppearbeid i digitalt klasserom
- Veiledning en-til-en og i gruppe
- Felles oppstart og morgenmøte
- Refleksjonslogg med skriftlige tilbakemeldinger
- Kundedialog
- Egenarbeid

**Veiledning og oppfølging:**

Det gis totalt 84 timer veiledning til studentene. Dette er fordelt med 12 timer veiledning pr. uke i 7 uker. Veiledningen er på oppgaver og arbeidskrav og gjennomføres etter avtale med studentene og fordeles på veiledning individuelt og i grupper der det er hensiktsmessig.

**Arbeidskrav og avsluttende emneeksamen for emne 2 IT-utvikling, prosess og metodikk**

1 obligatorisk oppgave skal løses og leveres inn i form av kildekode til case-oppgave laget i gruppe og et skriftlig dokument. Studenten skal drøfte og reflektere i lys av relevant faglitteratur. Oppgavens omfang: 1200 ord (10% +/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstr. 12, Times New Roman og 1.5 linjeavstand.

**Vurderingsform for arbeidskrav og avsluttende emneeksamen for emne 2 IT-utvikling, prosess og metodikk**

Arbeidskrav og avsluttende emneeksamen i form av en skriftlig innlevering vurderes med bestått/ikke bestått. Studenten får en skriftlig tilbakemelding samt bestått/ikke bestått innen 3 uker etter at innlevering ble mottatt. Karakteren bestått er emnekarakter som føres på vitnemålet.

**Vurderingskriterier for arbeidskrav og avsluttende emneeksamen for emne 2 IT-utvikling, prosess og metodikk**

Vurderingskriterier for arbeidskrav og avsluttende emneeksamen gis i forhold til 2-4 valgte læringsutbyttebeskrivelser i emne 1 Grunnleggende programmering i JavaScript på en nettside. Formelle krav: Oppgavens omfang: 600 ord (+/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstørrelse 12, Times New Roman og linjestørrelse 1.5. Besvarelsen leveres elektronisk innen fastsatt tid anonymt for sensor.

**Litteratur:**

ISBN 9788202661243 Koding for alle i JavaScript, Terje Kolderup, Cappelen Damm Akademisk, 2020

Kapittel 1 - 13, 176 sider

ISBN 978-3319051543 Agile!: The Good, the Hype and the Ugly, Bertrand Meyer, Springer, 2014

Kapittel 1 - 9, 144 sider

ISBN 9781119795780 Beginning C# and .NET, 2021 Edition, Benjamin Perkins, Jon D. Reid, Wiley, 2021

Kapittel 1 - 12, 354 sider

## Emne 3 C# og objektorientert programmering

### Kunnskaper:

#### Kandidaten

- har kunnskap om klasser, innkapsling og polymorfisme som anvendes innen C# og objektorientert programmering
- har kunnskap om objektorientert analyse og design som anvendes innen IT-utvikling
- har kunnskap om rike brukergrensesnitt som anvendes innen helhetlige programvareløsninger
- har kunnskap om enhetstesting som anvendes innen objektorientert programmering
- har kunnskap om kodeutviklingsmiljø som anvendes innen programmering i C#
- har kunnskap om refactoring som anvendes i objektorientert programmering
- har kunnskap om likheter og forskjeller mellom dynamisk og sterk typede språk, samt mellom tolkede og kompilerte språk som anvendes innen IT-utvikling
- kan oppdatere sin kunnskap om objektorientert programmering, Single Responsibility Principle og Dependency Inversion Principle gjennom informasjonsinnhenting og kontakt med fagmiljøet

### Ferdigheter:

#### Kandidaten

- kan anvende kunnskap om klasser, innkapsling og polymorfisme til objektorientert programmering og IT-utvikling
- kan anvende kunnskap om rike brukergrensesnitt til utvikling av helhetlige programvareløsninger
- kan anvende programmering, metodikk, prosess og verktøy til utvikling av programvare
- kan anvende refactorings-teknikkene innføre variabel, innføre funksjon og innføre klasse
- har anvende kunnskap om likheter og forskjeller mellom dynamisk og sterk typede språk, samt mellom tolkede og kompilerte språk i valg av teknologi i IT-utvikling
- kan anvende kodeutviklingsmiljø og versjonsstyringssystem for C# i utvikling av programvare
- kan anvende objektorientert analyse og design i IT-utvikling
- kan anvende enhetstesting til kontroll og kvalitetssikring av objektorientert programmering
- kan anvende refactoring i objektorientert programmering
- kan finne informasjon og fagstoff om objektorientert programmering, Single Responsibility Principle og Dependency Inversion Principle som er relevant for utvikling av programvare
- kan kartlegge status for eget arbeid innen objektorientert programmering og identifisere faglige problemstillinger og behov for endring i bruk av kildekode og teknologi

**Generell kompetanse:****Kandidaten**

- kan utføre objektorientert programmering ut fra behovet i et oppdrag
- kan bygge relasjoner med faggrupper innen IT-utvikling og på tvers av sektorer, samt med eksterne utviklere, designere og andre IT-medarbeidere
- kan bidra med objektorientert programmering som en del av helhetlige programvareløsninger

**Tema:**

- Sterk typing og kompilering
- Grunnleggende strukturert programmering i C#
- Input fra og output til fil
- Objekter av egne klasser, innkapsling
- Objektorientert analyse og design
- Oversette datastrukturer fra JavaScript til C#
- Enhetstesting av klasser
- Interface og arv
- Exceptions
- Single Responsibility Principle
- Dependency Inversion Principle
- Introduksjon til rike brukergrensesnitt og WPF

**Undervisningsformer og læringsaktiviteter:**

- Nettundervisning i digitalt klasserom
- Gruppearbeid i digitalt klasserom
- Veiledning en-til-en og i gruppe
- Felles oppstart og morgenmøte
- Refleksjonslogg med skriftlige tilbakemeldinger
- Egenarbeid

**Veiledning og oppfølging:**

Det gis totalt 72 timer veiledning til studentene. Dette er fordelt med 12 timer veiledning pr. uke i 6 uker. Veiledningen er på oppgaver og arbeidskrav og gjennomføres etter avtale med studentene og fordeles på veiledning individuelt og i grupper der det er hensiktsmessig.

**Arbeidskrav og avsluttende emneeksamen for emne 3 C# og objektorientert programmering**

1 obligatorisk oppgave skal løses og leveres inn i form av kildekode og et skriftlig dokument. Studenten skal drøfte og reflektere i lys av relevant faglitteratur. Oppgavens omfang: 600 ord (10% +/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstr. 12, Times New Roman og 1.5 linjeavstand.

**Vurderingsform for arbeidskrav og avsluttende emneeksamen for emne 3 C# og objektorientert programmering**

Arbeidskrav og avsluttende emneeksamen i form av en skriftlig innlevering vurderes med bestått/ikke bestått. Studenten får en skriftlig tilbakemelding samt bestått/ikke bestått innen 3 uker etter at innlevering ble mottatt. Karakteren bestått er emnekarakter som føres på vitnemålet.

### **Vurderingskriterier for arbeidskrav og avsluttende emneeksamen for emne 3 C# og objektorientert programmering**

Vurderingskriterier for arbeidskrav og avsluttende emneeksamen gis i forhold til 2-4 valgte læringsutbyttebeskrivelser i emne 1 Grunnleggende programmering i JavaScript på en nettside. Formelle krav: Oppgavens omfang: 600 ord (+/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstørrelse 12, Times New Roman og linjestørrelse 1.5. Besvarelsen leveres elektronisk innen fastsatt tid anonymt for sensor.

#### **Litteratur:**

ISBN 9788202661243 Koding for alle i JavaScript, Terje Kolderup, Cappelen Damm Akademisk, 2020

Kapittel 1 - 13, 176 sider

ISBN 978-3319051543 Agile!: The Good, the Hype and the Ugly, Bertrand Meyer, Springer, 2014

Kapittel 1 - 9, 144 sider

ISBN 9781119795780 Beginning C# and .NET, 2021 Edition, Benjamin Perkins, Jon D. Reid, Wiley, 2021

Kapittel 1 - 12, 354 sider



## 8 Beskrivelse av den indre sammenhengen i utdanningen \*

### NKR 5.1 Kunnskap:

#### Kandidaten

- har kunnskap om begreper, prosesser og verktøy som anvendes innenfor et spesialisert fagområde

### Overordnet læringsutbytte:

#### Kandidaten:

- har kunnskap om strukturert og objektorientert programmering, utviklingsmetodikk og -prosess, samt ulike verktøy som anvendes innen utvikling av programvare
- har kunnskap om etikk, personvern og GDPR som er relevant for IT-utvikling og programmering

### Emne 1 Grunnleggende programmering i JavaScript på en nettside

- har kunnskap om HTML, CSS og strukturert programmering, samt kodeutviklingsmiljø og versjonsstyringssystem som anvendes innen grunnleggende programmering i JavaScript på en nettside
- har kunnskap om design mønsteret model-view-controller og enhetstesting som anvendes innen utvikling av nettbasert programvare
- har kunnskap om konkrete problemløsningsteknikker som anvendes innen IT-utvikling og programmering
- har kunnskap om refactoring som anvendes innen strukturert programmering
- har kunnskap om etikk, personvern og GDPR som er relevant for programmering

### Emne 2 IT-utvikling, prosess og metodikk

- har kunnskap om systemutviklingsmetodikk som anvendes innen IT-utvikling
- har kunnskap om utviklingsprosessen som anvendes innen IT-utvikling
- har kunnskap om verktøy for prototypiering av skjermbilder som anvendes innen IT-utvikling
- har kunnskap om model-view-controller og objekter som anvendes ved datamodellering
- har kunnskap om etikk, personvern og GDPR som er relevant for IT-utvikling

### Emne 3 C# og objektorientert programmering

- har kunnskap om klasser, innkapsling og polymorfisme som anvendes innen C# og objektorientert programmering
- har kunnskap om objektorientert analyse og design som anvendes innen IT-utvikling
- har kunnskap om rike brukergrensesnitt som anvendes innen helhetlige programvareløsninger
- har kunnskap om enhetstesting som anvendes innen objektorientert programmering
- har kunnskap om kodeutviklingsmiljø som anvendes innen programmering i C#
- har kunnskap om refactoring som anvendes i objektorientert programmering
- har kunnskap om likheter og forskjeller mellom dynamisk og sterk typede språk, samt mellom tolkede og kompilerte språk som anvendes innen IT-utvikling

### NKR 5.1 Kunnskap:

#### Kandidaten

- har innsikt i relevant regelverk, standarder, avtaler og krav til kvalitet

### Overordnet læringsutbytte:

#### Kandidaten:

- har innsikt i relevant regelverk i forhold til kvalitetskrav og normer for programmering og IT-utvikling

### Emne 1 Grunnleggende programmering i JavaScript på en nettside

- har innsikt i relevant regelverk i forhold til kvalitetskrav og normer for grunnleggende programmering i JavaScript på en nettside

### Emne 2 IT-utvikling, prosess og metodikk

- har innsikt i relevant regelverk i forhold til kvalitetskrav og normer for programmering og IT-utvikling

**NKR 5.1 Kunnskap:****Kandidaten**

- har bransjekunnskap og kjennskap til yrkesfeltet

**Overordnet læringsutbytte:****Kandidaten:**

- har kunnskap om programmering og IT-utvikling og har kjennskap til utvikling av programvare for ulike yrkesfelt

**Emne 2 IT-utvikling, prosess og metodikk**

- har kunnskap om programmering og IT-utvikling og har kjennskap til utvikling av programvare for ulike yrkesfelt

**NKR 5.1 Kunnskap:****Kandidaten**

- kan oppdatere sin yrkesfaglige kunnskap

**Overordnet læringsutbytte:****Kandidaten:**

- kan oppdatere sin kunnskap om IT-utvikling gjennom informasjonsinnhenting om trender og teknologier og kontakt med fagmiljøet

**Emne 1 Grunnleggende programmering i JavaScript på en nettside**

- kan oppdatere sin kunnskap om grunnleggende programmering i JavaScript på en nettside gjennom informasjonsinnhenting om trender og teknologier og kontakt med fagmiljøet

**Emne 2 IT-utvikling, prosess og metodikk**

- kan oppdatere sin kunnskap om IT-utvikling gjennom informasjonsinnhenting om trender, teknologier, prosess og metodikk og kontakt med fagmiljøet

**Emne 3 C# og objektorientert programmering**

- kan oppdatere sin kunnskap om objektorientert programmering, Single Responsibility Principle og Dependency Inversion Principle gjennom informasjonsinnhenting og kontakt med fagmiljøet

**NKR 5.1 Kunnskap:****Kandidaten**

- forstår egen bransjes/yrkes betydning i et samfunns- og verdiskapingsperspektiv

**Overordnet læringsutbytte:****Kandidaten:**

- forstår betydningen av programmering og IT-utvikling i et samfunns- og verdiskapingsperspektiv

**Emne 2 IT-utvikling, prosess og metodikk**

- forstår betydningen av programmering og IT-utvikling i et samfunns- og verdiskapingsperspektiv

**NKR 5.1 Ferdigheter:****Kandidaten**

- kan anvende faglig kunnskap på praktiske og teoretiske problemstillinger

**Overordnet læringsutbytte:****Kandidaten:**

- kan anvende programmering, metodikk, prosess og verktøy til utvikling av programvare

**Emne 1 Grunnleggende programmering i JavaScript på en nettside**

- kan anvende koding i HTML og CSS og strukturert programmering til utvikling av grunnleggende programmering i JavaScript på en nettside
- kan anvende kunnskap om refactorings-teknikker i strukturert programmering
- kan anvende design mønsteret model-view-controller i egen programmering i JavaScript på en nettside

**Emne 2 IT-utvikling, prosess og metodikk**

- kan anvende kunnskap om systemutviklingsmetodikk til utvikling av programvare
- kan anvende kunnskap om utviklingsprosessen i utvikling av programvare
- kan anvende kunnskap om model-view-controller og objekter i datamodellering
- kan anvende kunnskap om etikk, personvern og GDPR som er relevant for IT-utvikling

**Emne 3 C# og objektorientert programmering**

- kan anvende kunnskap om klasser, innkapsling og polymorfisme til objektorientert programmering og IT-utvikling
- kan anvende kunnskap om rike brukergrensesnitt til utvikling av helhetlige programvareløsninger
- kan anvende programmering, metodikk, prosess og verktøy til utvikling av programvare
- kan anvende refactorings-teknikkene innføre variabel, innføre funksjon og innføre klasse
- har anvende kunnskap om likheter og forskjeller mellom dynamisk og sterk typede språk, samt mellom tolkede og kompilerte språk i valg av teknologi i IT-utvikling

**NKR 5.1 Ferdigheter:****Kandidaten**

- kan anvende relevante faglige verktøy, materialer, teknikker og uttrykksformer

**Overordnet læringsutbytte:****Kandidaten:**

- kan anvende kodeutviklingsmiljø og versjonsstyringssystem i utvikling av programvare

**Emne 1 Grunnleggende programmering i JavaScript på en nettside**

- kan anvende kodeutviklingsmiljø og versjonsstyringssystem i utvikling og feilsøking av programvare
- kan anvende problemløsningsteknikker til utvikling og feilretting av programvare
- kan anvende enhetstesting til kontroll og kvalitetssikring av programvare

**Emne 2 IT-utvikling, prosess og metodikk**

- kan anvende metodikk, prosess og verktøy til utvikling av programvare
- kan anvende verktøy for prototyping av skjermbilder i IT-utvikling

**Emne 3 C# og objektorientert programmering**

- kan anvende kodeutviklingsmiljø og versjonsstyringssystem for C# i utvikling av programvare
- kan anvende objektorientert analyse og design i IT-utvikling
- kan anvende enhetstesting til kontroll og kvalitetssikring av objektorientert programmering
- kan anvende refactoring i objektorientert programmering

**NKR 5.1 Ferdigheter:****Kandidaten:**

- kan finne informasjon og fagstoff som er relevant for en yrkesfaglig problemstilling

**Overordnet læringsutbytte:****Kandidaten:**

- kan finne informasjon og fagstoff om programmering og IT-utvikling som er relevant for en programvareløsning

**Emne 1 Grunnleggende programmering i JavaScript på en nettside**

- kan finne informasjon og fagstoff som er relevant for programmering i JavaScript på en nettside

**Emne 2 IT-utvikling, prosess og metodikk**

- kan finne informasjon om trender, teknologier, prosess og metodikk som er relevant for en programvareløsning

**Emne 3 C# og objektorientert programmering**

- kan finne informasjon og fagstoff om objektorientert programmering, Single Responsibility Principle og Dependency Inversion Principle som er relevant for utvikling av programvare

**NKR 5.1 Ferdigheter:****Kandidaten:**

- kan kartlegge en situasjon og identifisere faglige problemstillinger og behov for iverksetting av tiltak

**Overordnet læringsutbytte:****Kandidaten:**

- kan kartlegge status for eget arbeid innen IT-utvikling og programmering og identifisere faglige problemstillinger og behov for endring i bruk av kildekode og teknologi

**Emne 1 Grunnleggende programmering i JavaScript på en nettside**

- kan kartlegge status for eget arbeid innen programmering i JavaScript på en nettside og identifisere faglige problemstillinger og behov for endring i bruk av kildekode og teknologi

**Emne 2 IT-utvikling, prosess og metodikk**

- kan kartlegge et oppdrag fra en kunde og identifisere faglige problemstillinger og behov for prosess, metodikk, programmering og IT-utvikling

**Emne 3 C# og objektorientert programmering**

- kan kartlegge status for eget arbeid innen objektorientert programmering og identifisere faglige problemstillinger og behov for endring i bruk av kildekode og teknologi

**NKR 5.1 Generell kompetanse:****Kandidaten:**

- har forståelse for yrkes- og bransjeetiske prinsipper

**Overordnet læringsutbytte:****Kandidaten:**

- har forståelse for etiske prinsipper i IT-utviklingsbransjen

**Emne 1 Grunnleggende programmering i JavaScript på en nettside**

- har forståelse for etiske prinsipper i grunnleggende programmering i JavaScript på en nettside

**Emne 2 IT-utvikling, prosess og metodikk**

- har forståelse for etiske prinsipper i IT-utviklingsbransjen

**NKR 5.1 Generell kompetanse:****Kandidaten:**

- har utviklet en etisk grunnholdning i utøvelsen av yrket

**Overordnet læringsutbytte:****Kandidaten:**

- har utviklet en etisk grunnholdning i utøvelsen av programmering og IT-utvikling

**Emne 1 Grunnleggende programmering i JavaScript på en nettside**

- har utviklet en etisk grunnholdning i utøvelsen av grunnleggende programmering i JavaScript på en nettside

**Emne 2 IT-utvikling, prosess og metodikk**

- har utviklet en etisk grunnholdning i utøvelsen av programmering og IT-utvikling

**NKR 5.1 Generell kompetanse:****Kandidaten:**

- kan utføre arbeidet etter utvalgte målgruppers behov

**Overordnet læringsutbytte:****Kandidaten:**

- kan utføre utvikling av programvare etter utvalgte sektors behov

**Emne 1 Grunnleggende programmering i JavaScript på en nettside**

- kan utføre grunnleggende programmering i JavaScript på en nettside ut fra behovet i et oppdrag

**Emne 2 IT-utvikling, prosess og metodikk**

- kan utføre prosess, metodikk, programmering og IT-utvikling etter kundens behov

**Emne 3 C# og objektorientert programmering**

- kan utføre objektorientert programmering ut fra behovet i et oppdrag

**NKR 5.1 Generell kompetanse:****Kandidaten:**

- kan bygge relasjoner med fagfeller og på tvers av fag, samt med eksterne målgrupper

**Overordnet læringsutbytte:****Kandidaten:**

- kan bygge relasjoner med faggrupper innen IT-utvikling og på tvers av sektorer, samt med eksterne utviklere, designere og andre IT-medarbeidere

**Emne 1 Grunnleggende programmering i JavaScript på en nettside**

- kan bygge relasjoner med faggrupper innen grunnleggende programmering i JavaScript på en nettside og på tvers av sektorer, samt med eksterne utviklere, designere og andre IT-medarbeidere

**Emne 2 IT-utvikling, prosess og metodikk**

- kan bygge relasjoner med fagfeller og på tvers av fag, samt med kunder og brukere av programvaren

**Emne 3 C# og objektorientert programmering**

- kan bygge relasjoner med faggrupper innen IT-utvikling og på tvers av sektorer, samt med eksterne utviklere, designere og andre IT-medarbeidere

**NKR 5.1 Generell kompetanse:****Kandidaten:**

- kan utvikle arbeidsmetoder, produkter og/eller tjenester av relevans for yrkesutøvelsen

**Overordnet læringsutbytte:****Kandidaten:**

- kan bidra til IT-utvikling av helhetlige programvareløsninger

**Emne 1 Grunnleggende programmering i JavaScript på en nettside**

- kan bidra med grunnleggende programmering i JavaScript på en nettside som en del av helhetlige programvareløsninger

**Emne 2 IT-utvikling, prosess og metodikk**

- kan bidra med prosess, metodikk, programmering og IT-utvikling som en del av helhetlige programvareløsninger

**Emne 3 C# og objektorientert programmering**

- kan bidra med objektorientert programmering som en del av helhetlige programvareløsninger

## 9 Undervisningsformer og læringsaktiviteter

### Beskrivelse av undervisningsformer og læringsaktiviteter:

Alle studenter ved Fagskolen GET Academy AS følger samme progresjon. Læring er en prosess som krever aktiv innsats og kontinuerlig arbeid av studentene. De ulike læringsaktiviteter og undervisningsformer skal gi studentene trening i å søke kunnskap, kritisk tenkning og problemløsning. Variasjon i aktiviteter og oppgavetyper er nødvendig for at studentene skal oppnå en helhetlig kompetanse som omfatter både kunnskaper, ferdigheter og generell kompetanse.

Vi benytter følgende undervisningsformer og læringsaktiviteter:

- Nettundervisning i digitalt klasserom
  - Forelesning
  - Diskusjon
  - Spørsmål og svar
- Gruppearbeid i digitalt klasserom
- Parprogrammering
- Felles oppstart og morgenmøte med generiske ferdigheter
- Refleksjonslogg med skriftlige tilbakemeldinger
- Kundedialog
- Egenarbeid
  - Videoleksjoner
  - Øvingsoppgaver
  - Løsningsforslag i form av kildekode
  - Videoleksjon som forklarer løsningsforslag
  - Lese litteratur

### Kommunikasjons- og læringsplattform

Skolen benytter seg av Moodle som læringsplattform og Discord og Google som kommunikasjonsplattform. Av tjeneste fra Google bruker vi epost, deling av filer ved hjelp av Google Disk og Digitale møter via Google Meet.

Studentene vil få tilgang til og tilbud om opplæring i Moodle, Google og Discord. Det vil bli satt av tid til opplæring de to første ukene for å sikre at alle er fortrolig med bruken av Moodle, Google og Discord. På Discord kan det legges ut aktuell informasjon som timeplaner, leksjoner, oppgaver og henvisninger. Studentarbeid kan leveres inn via epost og Discord. Discord og epost gjør det også mulig for studenten å opprettholde kontakt med administrasjon, faglærer og medstudenter. Gjennom denne kommunikasjonen kan studenten bli veiledet og bevisstgjort slik at hun/han lettere kan ta i bruk kunnskap, ferdigheter og generell kompetanse i arbeid mot å oppnå læringsutbytte.

Læringsplattformen Moodle fungerer som studentenes klasserom og gir oversikt over studiet. Her finner studentene planer for hva som skal gås igjennom til enhver tid, læringsutbyttebeskrivelser, lærestoff og læringsaktiviteter. Læringsplattformen gir studentene også innsikt i studieplanen, læremidler, bakgrunnsstoff, oppgaver og diskusjonsforum.

Studentene får opplæring i bruk av skolens læringsplattform. Studentene får testet at eget utstyr fungerer som forutsatt. Studentene får introduksjon av utdanningens innhold og læringsutbytte ved studiestart. Det gis innføring i studieteknikk.

### **Nettundervisning i digitalt klasserom**

Nettundervisningen i digitalt klasserom foregår i Google Meet. Det er faglærer som er ansvarlig for nettundervisningen, og det er alltid to lærere til stede. Den ene underviser, og den andre har som oppgave å stille spørsmål og hjelpe til at gruppen fungerer best mulig.

Utdanningen legger vekt på undervisningsformer og læringsaktiviteter som fordrer aktiv deltakelse fra studentene, og vil ta utgangspunkt i studentenes egne erfaringer og problemstillinger, generelle utfordringer i arbeidslivet og teori.

Undervisningsformer og læringsaktivitetene vil variere i de ulike emnene, og tilrettelegges med tanke på å utvikle studentenes evne til samarbeid og økt forståelse og respekt for andres fagfelt, samt bidra til å styrke studentenes motivasjon for læring og utvikling. Studentene kan komme med innspill når det gjelder undervisningsformer og læringsaktiviteter. Undervisningsformene og læringsaktivitetene som benyttes skal bidra til å styrke studentenes motivasjon for utvikling og læring, og bidra til at de opplever nytte og gjensidighet gjennom faglig samarbeid. Variasjon i valg av metoder vil være viktig for å oppnå studiets overordnede læringsutbytte. Pedagogiske tilnærminger som, diskusjoner, oppgaver, gruppearbeid og forelesninger vil være naturlige variasjoner.

Det vil legges vekt på arbeid i grupper og på at studentene øver på tema og metoder som utdanningen berører. Studieplanen angir arbeidsformer i de ulike emnene og faglærer vil vurdere hvilke metoder som til enhver tid egner seg med tanke på tema og hvor de er i studieløpet.

Forelesning er en av formene for nettundervisning. Da går læreren gjennom fagstoff via eksempler og forklarer teorien, begrepene og programkoden som gjennomgås.

Diskusjon er en undervisningsform der lærer stiller spørsmål og fasiliterer en diskusjon mellom studentene.

Spørsmål og svar er en form for nettundervisning der læreren tar utgangspunkt i spørsmål som studentene har og besvarer disse med eksempler og forklaringer.

### **Gruppearbeid i digitalt klasserom**

I gruppearbeid består gruppene av 3 til 6 studenter. Faglærer setter sammen gruppene eller det kan gjøres i samarbeid med studentene. Ved gruppearbeid får studentene øvelse i å kommunisere med andre, gi og motta konstruktiv kritikk. Det vil legges vekt på arbeid i grupper og på at studentene øver på tema og metoder som utdanningen berører.

Innholdet i gruppearbeidet kan være egne case oppgaver gitt av faglærer, vurdering av hverandres arbeid, og det kan være samarbeid om læringsaktivitetene som også gjøres som egenarbeid, så som oppgaveløsning og gjennomgang av løsningsforslag.



Diskusjoner blir brukt for å utvide studentenes perspektiv og kunnskap. Gjennom åpen dialog deles synspunkter, meninger og erfaringer som kan være til nytte for studentene når det gjelder å uttrykke seg faglig.

### **Parprogrammering**

Parprogrammering er en egen metodikk i IT-utviklingsfaget, hvor den ene personen har rollen "sjåfør", hvilket vil si at denne personen sitter med mus og tastatur og skriver programkode. Den andre personen har rolle som observatør og navigatør. Denne personen skal til enhver tid vurdere om sjåføren er på riktig vei. Den kan komme med forslag til andre måter å løse problemet på, den kan være en sparringpartner, og den kan gjøre sjåføren klar over feil som denne har gjort. Parprogrammering brukes som en av mange teknikker i yrkesutøvelsen som IT-utvikler. Det er enighet om at parprogrammering produserer kode av høyere kvalitet, og at det har en klar pedagogisk effekt.

### **Refleksjonslogg med skriftlige tilbakemeldinger**

Studentene skriver i en refleksjonslogg hver dag. Her oppsummerer de faglig progresjon, utfordringer, motivasjon, samt kort sine egne perspektiv på dagens oppgave i generiske ferdigheter. Loggen leses av en lærer, og ved behov gis det skriftlig tilbakemelding og innspill. Ved hjelp av informasjon fra loggen, kan lærer gå inn tidlig dersom kursdeltaker har behov for ekstra oppfølging.

### **Kundedialog**

Å ha dialog med en kunde rundt et oppdrag er en viktig læringsaktivitet. Det går ut på å forstå et behov og foreslå en løsning, samt å ha tett dialog for å avdekke om man løser det rette problemet og om man løser det slik kunden ønsker. Det innebærer å gå gjennom det arbeidet som er gjort og be om tilbakemelding fra kunden.

### **Egenarbeid**

Egenarbeid er svært viktig i IT-utvikling og programmering. For å opparbeide seg ferdigheter i faget er det viktig å løse øvingsoppgaver. Læringsmateriellet ellers er viktig for å gi studentene nok kunnskap til å kunne gå i gang med løsning av øvingsoppgaver. Erfaringene fra løsning av øvingsoppgaver er viktig for å kunne tilegne seg forståelse av konseptene og begrepene i faget.

I tillegg til løsning av øvingsoppgaver innebærer egenarbeid gjennomgang av videoleksjoner både med forklaring av pensum og forklaring av løsningsforslag til øvingsoppgavene, samt lesing av litteratur. Egenarbeid er også eget arbeid med eksperimentering og uttesting av konsepter i pensum i egne prosjekter.

### **Undervisningsformer og læringsaktiviteter for emne 1 Grunnleggende programmering i JavaScript på en nettside**

- Nettundervisning i digitalt klasserom
- Gruppearbeid i digitalt klasserom
- Veiledning en-til-en og i gruppe
- Felles oppstart og morgenmøte
- Refleksjonslogg med skriftlige tilbakemeldinger
- Egenarbeid

## **Undervisningsformer og læringsaktiviteter for emne 2 IT-utvikling, prosess og metodikk**

- Nettundervisning i digitalt klasserom
- Gruppearbeid i digitalt klasserom
- Veiledning en-til-en og i gruppe
- Felles oppstart og morgenmøte
- Refleksjonslogg med skriftlige tilbakemeldinger
- Kundedialog
- Egenarbeid

## **Undervisningsformer og læringsaktiviteter for emne 3 C# og objektorientert programmering**

- Nettundervisning i digitalt klasserom
- Gruppearbeid i digitalt klasserom
- Veiledning en-til-en og i gruppe
- Felles oppstart og morgenmøte
- Refleksjonslogg med skriftlige tilbakemeldinger
- Egenarbeid

## **Pedagogisk opplegg**

Skolens pedagogiske modell baserer seg på nettundervisning i digitalt klasserom hver uke, egenarbeid for studenten individuelt og i grupper, samt ukentlig veiledning av studenter både en-til-en og i grupper.

Skolen legger mye innsats i å bygge et miljø med høy grad av psykologisk trygghet for studentene. En sentral del av det pedagogiske opplegget er det vi kaller generiske ferdigheter, og vi sorterer dem i tre hovedkategorier, med hver sine undertemaer:

1. tenke og lære
  - a. etisk refleksjon
  - b. kreativ problemløsning
  - c. kritisk tenkning
  - d. metakognisjon
  - e. læringsstrategier
2. samarbeid og kommunikasjon
  - a. aktiv lytting
  - b. psykologisk trygghet
  - c. sosial kompetanse
  - d. tuckmans modell rundt teamdynamikk
3. selvledelse og refleksjon
  - a. egenmotivasjon og prokrastinering
  - b. emosjonell kompetanse - forstå, uttrykke og regulere egne følelser
  - c. gode vaner
  - d. grit
  - e. growth mindset
  - f. mestringspsykologi

## g. personlighetspsykologi

Regjeringen bruker begrepet generiske ferdigheter i “NOU 2020: 2 Fremtidige kompetansebehov III”, og der er det definert som “en rekke ferdigheter ervervet gjennom høyere utdanning, som ikke er direkte knyttet til et fag, disiplin eller yrke som høyere utdanning utdanner til”. Fire dager i uken har vi undervisning om ukens tema innen generiske ferdigheter. Deretter jobber studentene i grupper, med erfaringsdeling og refleksjon. De får oppgaver knyttet til dagens tema - som de reflekterer over sammen ut fra egne erfaringer, ressurser og utfordringer, i den lærings situasjonen de står i, samt ut fra fremtidig yrkesutøvelse.

Det pedagogiske opplegget er videre basert på at studentene gjør mye praktisk programmering. Vi prioriterer høyt å trene opp reelle ferdigheter i programmering. Dette innebærer løsning av øvingsoppgaver og egne prosjekter, og vi kan kalle det pedagogiske opplegget som “learning by doing”.

Øvingsoppgavene er viktige da de guider studentene gjennom fagstoffet på en praktisk måte hvor studentene samtidig er aktive. Innenfor hvert tema finnes det oppgaver på ulike nivåer. På det enkleste nivået handler det om å kunne oversette en detaljert tekstlig bestilling til programkode. De første oppgavene om et tema har ofte detaljerte beskrivelser i oppgaveteksten, mens senere oppgaver baserer seg på at studentene kan være mer selvstendige og at de klarer å gjenbruke mønsteret fra tidligere oppgaver sammen med kunnskapen fra lærestoffet.

Lærestoffet introduserer nye temaer, slik at studentene har nok kunnskap til å gjøre øvingsoppgavene. Lærestoffet forklarer begreper og konsepter på en praktisk måte gjennom konkrete kodeeksempler. Gjennom øvingsoppgavene får studentene erfaring som øker deres evne til å bygge forståelse. Lærestoffet forklarer faget ut fra den praktiske programmeringen studentene har vært gjennom så langt. Dette skjer gjennom de ulike læringsaktivitetene, så som videoleksjoner, nettundervisning og arbeid i grupper.

Opplegget inneholder også mye jobbing i gruppe, hvor man må snakke og skrive om faget. Å lære bort til andre er bra for egen læring, og dette utvikler også studentenes evne til å formulere faglige problemstillinger i muntlig og skriftlig språk.

### **Veiledning og oppfølging**

Veiledning gjennomføres etter avtale med studentene og fordeles på veiledning individuelt og i grupper der det er hensiktsmessig. Veiledning foregår via nettundervisningen i digitalt klasserom, via Discord, epost og telefon. Veiledningen gis til studentene i arbeid med oppgaver og arbeidskrav.

Tid som er satt av til veiledning vil avhenge av individuelle behov og gruppestørrelse. Det er estimert tid for veiledning i hvert emne. Hvis en student står i fare for å ikke oppnå lærings-utbyttebeskrivelsene i ett emne, skal faglærer så tidlig som mulig fange opp dette og i dialog med studenten legge en plan for videre arbeid. Planen skal beskrive både faglærers og studentens ansvar, samt klargjøre oppgaver, mål og tidsperspektiv. Planen skal være individuelt tilpasset den enkelte students behov. Studentenes behov for oppfølging og kontakt med lærer vil være tilpasset den enkelte students behov. Studentene vil få svar på sine spørsmål innen normalt 24 timer på virkedager. Studentene vil motta tilbakemeldinger på obligatoriske oppgaver

senest fem virkedager etter leveringsfrist, og for arbeidskrav er det innen senest 3 uker fra leveringsfristen. Veiledning og oppfølging foregår i det digitale klasserommet enten for gruppen samlet eller som individuell oppfølging. Veiledning og oppfølging er basert på at studentene skal oppnå læringsutbyttene og alle tilbakemeldingene, skriftlige eller muntlige, må derfor virke læringsfremmende.

Det utarbeides en plan for veiledning i første uke av studiet og denne justeres underveis i samarbeid med studentene etter hvert som det er nødvendig.

### **Emne 1 Grunnleggende programmering i JavaScript på en nettside**

Det gis totalt 84 timer veiledning til studentene. Dette er fordelt med 12 timer veiledning pr. uke i 7 uker. Veiledningen er på oppgaver og arbeidskrav og gjennomføres etter avtale med studentene og fordeles på veiledning individuelt og i grupper der det er hensiktsmessig.

### **Emne 2 IT-utvikling, prosess og metodikk**

Det gis totalt 84 timer veiledning til studentene. Dette er fordelt med 12 timer veiledning pr. uke i 7 uker. Veiledningen er på oppgaver og arbeidskrav og gjennomføres etter avtale med studentene og fordeles på veiledning individuelt og i grupper der det er hensiktsmessig.

### **Emne 3 C# og objektorientert programmering**

Det gis totalt 72 timer veiledning til studentene. Dette er fordelt med 12 timer veiledning pr. uke i 6 uker. Veiledningen er på oppgaver og arbeidskrav og gjennomføres etter avtale med studentene og fordeles på veiledning individuelt og i grupper der det er hensiktsmessig.

## 10 Arbeidskrav, emneeksamen og vurdering

**Arbeidskrav i hvert emne er avsluttende emneeksamen.**

### **Arbeidskrav og avsluttende emneeksamen for emne 1 Grunnleggende programmering i JavaScript på en nettside**

1 obligatorisk oppgave skal løses og leveres inn i form av kildekode og et skriftlig dokument. Studenten skal drøfte og reflektere i lys av relevant faglitteratur. Oppgavens omfang: 600 ord (10% +/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstr. 12, Times New Roman og 1.5 linjeavstand.

### **Arbeidskrav og avsluttende emneeksamen for emne 2 IT-utvikling, prosess og metodikk**

1 obligatorisk oppgave skal løses og leveres inn i form av kildekode til case-oppgave laget i gruppe og et skriftlig dokument. Studenten skal drøfte og reflektere i lys av relevant faglitteratur. Oppgavens omfang: 1200 ord (10% +/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstr. 12, Times New Roman og 1.5 linjeavstand.

### **Arbeidskrav og avsluttende emneeksamen for emne 3 C# og objektorientert programmering**

1 obligatorisk oppgave skal løses og leveres inn i form av kildekode og et skriftlig dokument. Studenten skal drøfte og reflektere i lys av relevant faglitteratur. Oppgavens omfang: 600 ord (10% +/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstr. 12, Times New Roman og 1.5 linjeavstand.

### **Vurdering**

Vurderingsformene skal ivareta sammenhengen med utdanningens læringsutbytte og læringsutbyttebeskrivelsene på emnenivå i tillegg til arbeidsmåter og læringsformer. Vurderingen skal foregå på en slik måte at studentene gis mulighet til å utvikle seg gjennom studiet, og derved erfare at kompetanse i et emne kan suppleres av kompetanse i et annet emne. Presise og relevante tilbakemeldinger er viktige i denne sammenhengen.

### **Vurderingsform for arbeidskrav og avsluttende emneeksamen for emne 1 Grunnleggende programmering i JavaScript på en nettside**

Arbeidskrav og avsluttende emneeksamen i form av en skriftlig innlevering vurderes med bestått/ikke bestått. Studenten får en skriftlig tilbakemelding samt bestått/ikke bestått innen 3 uker etter at innlevering ble mottatt. Karakteren bestått er emnekarakter som føres på vitnemålet.

### **Vurderingsform for arbeidskrav og avsluttende emneeksamen for emne 2 IT-utvikling, prosess og metodikk**

Arbeidskrav og avsluttende emneeksamen i form av en skriftlig innlevering vurderes med bestått/ikke bestått. Studenten får en skriftlig tilbakemelding samt bestått/ikke bestått innen 3 uker etter at innlevering ble mottatt. Karakteren bestått er emnekarakter som føres på vitnemålet.

### **Vurderingsform for arbeidskrav og avsluttende emneeksamen for emne 3 C# og objektorientert programmering**

Arbeidskrav og avsluttende emneeksamen i form av en skriftlig innlevering vurderes med bestått/ikke bestått. Studenten får en skriftlig tilbakemelding samt bestått/ikke bestått innen 3 uker etter at innlevering ble mottatt. Karakteren bestått er emnekarakter som føres på vitnemålet.

### **Vurderingskriterier for arbeidskrav og avsluttende emneeksamen for emne 1 Grunnleggende programmering i JavaScript på en nettside**

Vurderingskriterier for arbeidskrav og avsluttende emneeksamen gis i forhold til 2-4 valgte læringsutbyttebeskrivelser i emne 1 Grunnleggende programmering i JavaScript på en nettside. Formelle krav: Oppgavens omfang: 600 ord (+/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstørrelse 12, Times New Roman og linjestørrelse 1.5. Besvarelsen leveres elektronisk innen fastsatt tid anonymt for sensor.

### **Vurderingskriterier for arbeidskrav og avsluttende emneeksamen for emne 2 IT-utvikling, prosess og metodikk**

Vurderingskriterier for arbeidskrav og avsluttende emneeksamen gis i forhold til 2-4 valgte læringsutbyttebeskrivelser i emne 1 Grunnleggende programmering i JavaScript på en nettside. Formelle krav: Oppgavens omfang: 600 ord (+/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstørrelse 12, Times New Roman og linjestørrelse 1.5. Besvarelsen leveres elektronisk innen fastsatt tid anonymt for sensor.

### **Vurderingskriterier for arbeidskrav og avsluttende emneeksamen for emne 3 C# og objektorientert programmering**

Vurderingskriterier for arbeidskrav og avsluttende emneeksamen gis i forhold til 2-4 valgte læringsutbyttebeskrivelser i emne 1 Grunnleggende programmering i JavaScript på en nettside. Formelle krav: Oppgavens omfang: 600 ord (+/-). I oppgavens lengde er ikke forside, innholdsfortegnelse og referanseliste inkludert. Det forutsettes at det brukes punktstørrelse 12, Times New Roman og linjestørrelse 1.5. Besvarelsen leveres elektronisk innen fastsatt tid anonymt for sensor.

## **11 Litteraturliste/læremidler**

ISBN 9788202661243 Koding for alle i JavaScript, Terje Kolderup, Cappelen Damm Akademisk, 2020  
Kapittel 1 - 13, 176 sider

ISBN 978-3319051543 Agile!: The Good, the Hype and the Ugly, Bertrand Meyer, Springer, 2014  
Kapittel 1 - 9, 144 sider

ISBN 9781119795780 Beginning C# and .NET, 2021 Edition, Benjamin Perkins, Jon D. Reid, Wiley, 2021  
Kapittel 1 - 12, 354 sider

## Vedlegg 1: Undervisningsplan

Utdanningen gjennomføres over 800 timer som er fordelt med 40 timer gjennom 20 uker.

Uke	Tema	Pedagogisk opplegg	Undervisning per uke
Emne 1 Grunnleggende programmering i JavaScript på en nettside			
1	HTML, CSS, GIT og GitHub.com. Funksjoner i JavaScript - og endring av HTML-dokument fra JavaScript	Psykologisk trygghet	7 timer Nettundervisning i digitalt klasserom  6 timer Gruppearbeid i digitalt klasserom  2 timer Parprogrammering  4 timer Felles oppstart og morgenmøte med generiske ferdigheter  2 timer Refleksjonslogg med skriftlige tilbakemeldinger  19 timer Egenarbeid
2	Document Object Model. Funksjoner med parametre og returverdi. Variabler, tilordning og aritmetiske operatører. Refactoring.	Samarbeid og teamjobbing	5 timer Nettundervisning i digitalt klasserom
3	Design mønsteret model-view-controller. Valgsetninger. Sammenligningsoperatører og logiske operatører.	Læringsstrategier	4 timer Gruppearbeid i digitalt klasserom  2 timer Parprogrammering
4	Enhetstesting av funksjoner	Problem-løsning	4 timer Felles oppstart og morgenmøte med generiske ferdigheter
5	Lister og løkker	Kritisk tenkning	
6	Objekter	Gode vaner	2 timer Refleksjonslogg med skriftlige tilbakemeldinger
7	Objekter	Gode vaner	23 timer Egenarbeid
Emne 2 IT-utvikling, prosesser og metodikk			

8	Utviklingsprosessen. Utviklingsmetodikk. Kundedialog. Objekter, datastrukturer og modellering. Analyse og design. Tegning av skjermbilder.	Aktiv lytting, kommunikasjon	6 timer Nettundervisning i digitalt klasserom 8 timer Gruppearbeid i digitalt klasserom 4 timer Felles oppstart og morgenmøte med generiske ferdigheter 2 timer Refleksjonslogg med skriftlige tilbakemeldinger 20 timer Egenarbeid
9	Case-oppgave Systemutviklingsmetodikk	Sosial kompetanse	1 time Kundedialog 1 time Nettundervisning i digitalt klasserom
10	Case-oppgave Smidige metoder	Tuckmans modell rundt team-dynamikk	8 timer Gruppearbeid i digitalt klasserom
11	Case-oppgave Etikk	Etisk refleksjon	4 timer Felles oppstart og morgenmøte med generiske ferdigheter
12	Case-oppgave Personvern – GDPR	Emosjonell kompetanse - forstå, uttrykke og regulere egne følelser	2 timer Refleksjonslogg med skriftlige tilbakemeldinger
13	Case-oppgave Bærekraft	Growth mindset	24 timer Egenarbeid
14	Case-oppgave	Gode vaner	
Emne 3 C# og objektorientert programmering			
15	Strukturert programmering i C#. Sterk typing. Kompilering. Input fra og output til fil.	GRIT	5 timer Nettundervisning i digitalt klasserom
16	Objekter av egne klasser, innkapsling. Oversette datastrukturer fra JavaScript til C#. Exceptions.	Egenmotivasjon og prokrastinering	4 timer Gruppearbeid i digitalt klasserom 2 timer Parprogrammering
17	Objektorientert analyse og design. Enhetstesting av klasser.	Gode vaner	4 timer Felles oppstart og morgenmøte med generiske ferdigheter
18	Polymorfisme ved hjelp av interface og arv.	Personlighetspsykologi	



19	Introduksjon til rike brukergrensesnitt og WPF.	Metakognisjon	2 timer Refleksjonslogg med skriftlige tilbakemeldinger  23 timer Egenarbeid
20	Single Responsibility Principle. Dependency Inversion Principle.	Mestrings-psykologi	