Pedagogisk mappe

Henrik Hillestad Løvold

Juli 2025

1 Undervisningsfilosofi

Da jeg selv studerte, foretrakk jeg tradisjonelle forelesninger som undervisningsform. Det er behagelig å møte opp i et auditorium og høre på noen snakke om et interessant tema i et par timer. Men lærte jeg noe av det? Og hvor behagelig skal det egentlig være å lære?

Å lære noe i realfag generelt krever at man er i stand til å kjenne igjen mønstre. Når man lærer matematikk på skolen går mesteparten av tiden med til å regne, og tavleundervisning er en relativt liten del av undervisningen. I så måte minner informatikk, og programmering spesielt, mye om matematikk. Det er lett å sitte i et auditorium og følge forelesers resonnementer, uten å kunne anvende ferdighetene som foreleses på egenhånd i etterkant.

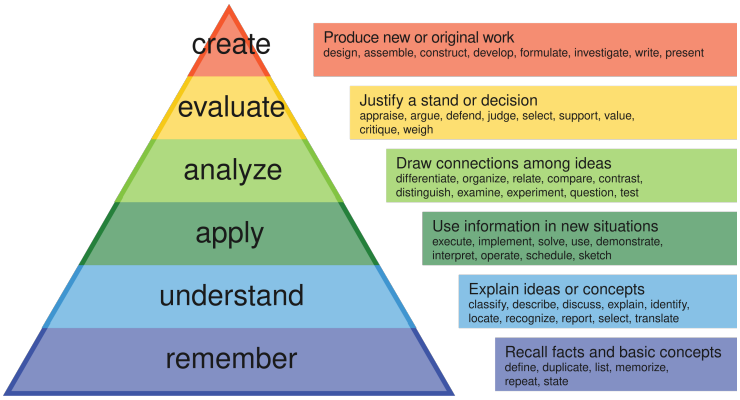
Disse observasjonene har ledet meg i retning av mer aktive undervisningsformer. Det er vanskelig å definere hva begrepet "aktiv undervisning" egentlig innebærer. For meg personlig handler det om all undervisning som ikke er styrt i én retning; i en "aktiv undervisningssammenheng" er det et samspill mellom underviser og student, til forskjell fra blant annet tradisjonelle forelesninger, hvor foreleser snakker til en forsamling. Enten det er snakk om problembasert læring, flipped classroom eller hverandrevurderinger, vil studenten aktivt ta del i læringssituasjonen.

Ser man til Blooms reviderte taksonomi (Krathwohl, 2002), som vist i figur 1[1], kan man si at målet med aktiv læring er å få studentene fra de lavere læringsnivåene, slik som å huske og forstå, til de høyere læringsnivåene slik som å anvende, analysere, evaluere og å skape på egen hånd.

Å involvere studentene i undervisningen er likevel ikke enkelt. Oppgavene som gis og diskusjonene som initieres må spisses, og de må adressere aktuelle temaer fra pensum. Vel så viktig som spørsmålet *hva er det jeg ønsker at studentene skal lære*, er spørsmålet *hvilke temaer er det studentene synes er spesielt vanskelige*. Å identifisere disse temaene er ikke nødvendigvis enkelt. Jeg har selv erfart at det er lenge siden jeg selv var student og lærte det jeg nå underviser, og det

[1] Illustrasjon ved brukeren Tidema, utgitt på lisens CC BY 4.0 på Wikimedia Commons.

er ikke alltid det jeg opplever som banalt, er banalt for studentene mine. På samme måte har jeg også tatt meg selv i å tenke at noe er virkelig vanskelig, som studentene ikke har funnet vanskelig i det hele tatt.



Figur 1: Blooms taksonomi, etter revisjon (Krathwohl, 2002).

I min rolle som universitetslektor har jeg gjort flere forsøk på å innlemme aktive komponenter i min undervisning. Noen ganger har det fungert godt, og andre ganger har det ikke fungert like godt. Jeg vil i dette kapittelet gjøre rede for hvilke undervisningsmetoder jeg har forsøkt, samt mine refleksjoner over disse, med et spesielt blikk på aktiv undervisning. Jeg vil også diskutere rollen som veileder og fagbokforfatter.

1.1 Problembasert læring

I 2021 og 2022 fikk jeg øynene opp for undervisningsformen *problembasert læring*. Denne formen for undervisning innebærer at studentene presenteres for et problem, før de selv gruppevis går i gang med å løse det, uten på forhånd å ha fått aktuelle konsepter presentert eller forelest av underviseren. Sentralt i problembasert læring, er at studentene selv skal forstå og løse problemet, uten å få direkte svar fra underviseren. Underviserens rolle blir mer en veilederrolle, enn en tradisjonell underviserrolle der man forteller studentene hva som er riktig og hvorfor. Målet med problembasert læring er i stor grad at studentene skal gjennomgå en utvikling fra bare å kunne gjengi kunnskap til å bli i stand til å anvende og analysere kunnskapsstoffet og evaluere eget læringsutbytte. Ser man til Blooms taksonomi (Krathwohl, 2002), handler problembasert læring om å få studentene til å klatre høyere i læringshierarkiet. Opprinnelig ble undervisningsmetoden utviklet av undervisere i medisin (Wood, 2003), men den benyttes nå innen en rekke ulike fagfelt, inkludert informatikk.

I arbeidet mitt ved UiT har jeg gradvis beveget meg mer og mer i retning av å bruke problembasert læring som undervisningsform. Jeg tok over kurset *INF-1049 – Introduksjon til beregningsorientert programmering* i 2021, som på

den tiden var et tradisjonelt forelesningsbasert kurs. Første gjennomføring av kurset som rent problembasert uten tradisjonelle forelesninger, var i 2024. Denne prosessen er mer detaljert beskrevet i kapittel 2.

Mine observasjoner fra undervisning med problembasert læring som metode, indikerer at studentene *lærer mer*, samtidig som de *føler at de lærer mindre*. Paradoksalt nok har karaktersnittet i kurset hvor jeg benyttet metoden gått opp, samtidig som studentenes rapporterte tilfredsstillelse med undervisningen har gått ned. For meg som underviser er selvfølgelig det viktigste at studentene lærer så mye som mulig i løpet av semesteret. Jeg har likevel vært nødt til å ta innover meg at studentenes opplevelse av læring og undervisningen de får ved universitetet, er viktig for både studentmiljøet og for frafallsprosenten fra studieprogrammene.

Da jeg i 2024 gjennomførte ren problembasert læring, var frustrasjonen hos studentene vanskelig å håndtere. Jeg hørte ofte "*du* vet jo svaret, hvorfor kan du ikke fortelle meg det?" I stedet for å fortelle studentene hva de hadde gjort feil, eller hvorfor de var på villspor, var jeg nødt til å be dem stille meg "hvorfor-spørsmål". Ikke nødvendigvis for at studentene skulle få svar på spørsmålet de stilte, men heller for å få dem til å reflektere over *hva* de ikke skjønte og *hvorfor* de satt fast. Å lære studentene å stille gode "hvorfor-spørsmål" er nok vel så viktig som å lære dem om konkrete konsepter – å være student ved et universitet handler i stor grad om å forstå hva man *ikke* forstår, og ikke minst hvordan man tilegner seg kunnskap på egen hånd.

Jeg har siden reflektert over konteksten jeg gjennomførte mitt forsøk med problembasert læring i. Kurset jeg brukte for forsøket er et førstesemester-kurs for bachelorstudenter, og det er nok mulig at de ikke enda er modne nok til å forstå min motivasjon ved å la være å bruke tradisjonelle forelesninger, og heller kun gi dem problemer og "ja, hvorfor det?"-svar på spørsmål. Det er ikke sikkert at denne undervisningsmetoden er egnet for bruk fra aller første undervisningsdag i første semester, men heller senere i studiet, eller i alle fall senere i semesteret. I det pågående porteføljearbeidet ved Institutt for informatikk, som er nærmere beskrevet i kapittel 2.5, planlegges det å introdusere problembasert læring som undervisningsform i siste halvdel av første semester. Da vil studentene allerede ha litt kunnskap å trekke linjer til når de skal gå i gang med å løse problemer uten direkte involvering fra faglærer.

1.2 Arbeid i team

Når studenter lærer seg å programmere for første gang, bygger de opp mentale modeller av hvordan ulike konsepter og konstruksjoner i et programmeringsspråk fungerer (Rørnes mfl., 2019). Disse mentale modellene danner et grunnlag for hvordan de forstår programmering som helhet, og det er min erfaring at misforståelser bør oppklares tidlig, slik at gale mentale modeller ikke får utvikle seg videre.

Selv om foreleser til en viss grad kan påvirke hvordan studenter tenker på kon-

septer innen programmering, er min erfaring at studentene lærer mer ved diskusjon med hverandre. Ved å eksponeres for andres forståelse, blir studentene klar over feil i egne modeller av konsepter. Diskusjon studenter imellom kan også være nyttig for studenter som allerede har dannet seg korrekt forståelse av stoffet, ettersom de får muligheten til å argumentere for egen tankegang, og dermed også bli klar over hvordan konsepter kan misforstås.

Da jeg arbeidet ved Universitetet i Oslo var jeg med på å utvikle kurset *IN2000 – Software Engineering med prosjektarbeid*. Dette kurset veier 20 studiepoeng og følges av studenter i sitt fjerde semester i bachelorstudiet. Kurset dreier seg i stor grad om prosjektarbeid, der studenter i team bestående av 4-6 personer utvikler en app. Vel så viktig som produktet er prosessarbeid i team, hvor studentene eksperimenterer med ulik utviklingsmetodikk. Spesielt for kurset er at det følges av bachelorstudenter fra tre ulike studieprogram; programmering og systemarkitektur, interaksjonsdesign og digital økonomi og ledelse. Dette gir mulighet for teamsammensetninger der deltakerne innad i hvert team sitter på ulik kompetanse fra tidligere semestre.

Etter at kurset gikk for første gang våren 2018 ble jeg og øvrige faglærere bekymret for at teamene, som studentene da hadde satt sammen selv, ble for homogene med tanke på studieprogram, kjønn og arbeidskapasitet. Som et eksperiment lot vi i 2019 studentene velge om de selv ville sette sammen team, eller om de ville settes sammen med andre studenter av faglærer basert på et nettskjema. Ved sammensetting av teamene tok vi høyde for studentenes studieprogram, kjønn og ambisjonsnivå, og forsøkte å sette sammen så heterogene team som mulig. Kun 22% av studentene ønsket å bli plassert i team av faglærerne, noe som resulterte i 9 team på fem til seks studenter. Vi fant at studenter som selv valgte team gjorde det noe bedre enn teamene vi hadde satt sammen. Våre funn resulterte i en konferanseartikkel i 2020 (Løvold mfl., 2020).

1.3 Peer instruction

For å måle studenters forståelse underveis i en tradisjonell forelesning, utviklet professor i fysikk Eric Mazur på 90-tallet et verktøy kalt *peer instruction* (Mazur & Somers, 1999). Underveis i forelesningen får studentene presentert et problem som skal besvares individuelt ved bruk av "mentometerknapper". Etterpå diskuteres svarene studentene imellom, og besvares på nytt. Når svarene etter diskusjon er avgitt, går foreleser igjennom problemet og de ulike svaralternativene. Vel så viktig som hvorfor ett er riktig, er det også å begrunne hvorfor de øvrige alternativene er gale. På denne måten gir man rom for å rette opp i vanlige misoppfatninger, om spørsmålene er designet på en god måte. Bruk av denne metoden gir også foreleser bedre grunnlag for å planlegge fremtidig undervisning, ettersom studentenes svar gir en indikasjon på hva som krever grundigere gjennomgang.

Selv om peer instruction opprinnelig ble brukt i undervisning i fysikk, brukes metoden aktivt også i andre fagfelt. Zingaro og Porter, 2015 har vist at metoden

også kan anvendes med hell i undervisning i programmering. I tillegg til de tradisjonelle elementene i peer instruction, ble det i deres studie lagt til et nytt element kalt *isomorphic questions*, eller isomorfe spørsmål. Dette er spørsmål som stilles til slutt, og som i teorien er helt like som det opprinnelige spørsmålet, men med en annen "innpakning".

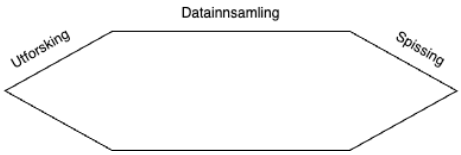
Jeg har selv gjort forsøk med peer instruction i forbindelse med forelesninger i kurset *IN1000 – Objektorientert programmering* ved Universitetet i Oslo høsten 2018. Min erfaring er at diskusjon mellom studenter fører til mange flere korrekte svar når spørsmålet stilles for andre gang. Hvorvidt dette skyldes den faktiske diskusjonen er dog noe uklart. Det kan godt tenkes at dyktige studenter kun forteller svakere studenter hva som er det korrekte svaret, uten å gi en god forklaring på hvorfor. Her kommer isomorfe spørsmål med individuell besvarelse inn som en god måte å måle forståelse på. Om enn ikke forskningsmessig signifikant, kan mine resultater til en viss grad indikere at studentene svarer mer korrekt på et tilsvarende problem etter diskusjon med medstudenter.

1.4 Veilederrollen

For studenter som skal skrive en masteroppgave er det et stort steg å skulle gå fra å følge emner som formidler forskningsbasert kunnskap, til selv å skulle gjennomføre en vitenskapelig studie. Å veilede studenter gjennom deres første forskningsprosjekt er et ansvar som krever struktur og bevissthet. Selv om man har en overskrift og en prosjektbeskrivelse, er ikke studenten nødvendigvis sikker på hva de faktisk ønsker å undersøke, og ikke minst hvordan man gjennomfører en vitenskapelig studie.

Studenter som skal begynne å arbeide med en oppgave har gjerne mye generell kunnskap fra tidligere i studiet sitt, men mangler inngående kunnskap i feltet de skal arbeide med i forbindelse med oppgaven sin. Når jeg har første møte med en student jeg skal veilede, benytter jeg ofte heksagonmodellen (Figur 2). Denne modellen fremstiller arbeidet med en studie som tre faser, fra venstre mot høyre. I starten har man et utgangspunkt, ofte en tittel og en oppgavebeskrivelse, og man starter med å utforske feltet. I denne fasen får studenten mer kjennskap til tidligere forskning, bakgrunn og kunnskap om domenet oppgaven omhandler. Ved slutten av denne fasen har studenten kjennskap til feltet, og er bedre i stand til å sette sin oppgave inn i en vitenskapelig kontekst. Under fase to, datainnsamling, gjennomfører studenten studien sin. Dette er en åpen fase hvor det er mulig å gjøre justeringer underveis, etter hvert som ny kunnskap blir etablert. Avslutningsvis, i fase tre, snevrer man igjen inn skopet og spisser oppgaven mot et ferdig arbeid.

Når jeg veileder masterstudenter tilbyr jeg ukentlige møter om studentene ønsker dette, og ved oppstarten avtaler vi faste tidspunkt for veiledningsmøtene. Selv om ikke alle studenter ønsker så hyppige møter, mener jeg at dette er nyttig for å følge progresjonen tett. Ved å ha ukentlige møter kan man løse opp i misforståelser tidlig, og det er et godt forum for å diskutere endringer siden



Figur 2: Heksagonmodellen.

sist. Like viktig er det at veileder fort kan oppdage skrivesperrer og mangelfull progresjon hos studenten. I slike tilfeller kan råd om hvordan man kommer seg videre være nyttige og hindre at studenten havner bakpå i arbeidet.

Veileders rolle er å lede studenten på rett vei, og ikke å styre prosjektet. Prosjektets eier vil alltid være studenten, og om studenten ønsker å gå i en annen retning enn den veien veilederen peker, så står de fritt til det. Dette kan i noen tilfeller skape utfordringer med videre veiledning, noe avhengig av studentens selvstendighet. Som veileder ser jeg det som del av rollen også å justere meg slik at jeg kan hjelpe studenten der de står, selv om det kan være vanskelig å gi råd når studenten har valgt en annen vei enn den jeg hadde sett for meg.

Å veilede en student handler ikke bare om den konkrete oppgaven studenten arbeider med. Det handler også om å gi studenten en arbeidsmetode og disiplin som kan brukes siden, enten det er i akademisk arbeid eller i annet arbeid etter endt studium. I starten av veiledningssituasjonen bruker jeg derfor gjerne en del tid på å gi råd om prosess. Kildekritikk er et annet moment som ikke er direkte knyttet til oppgaven, men som er en viktig kunnskap å ha med seg inn i et vitenskapelig arbeid. Også dette bruker jeg gjerne tid til å snakke om med studenten.

1.5 Fagbokforfatters rolle

Informatikk er et fagfelt som utvikler seg fort. Det som var relevant for fem år siden, kan være helt irrelevant i dag. Med nye teknologier kommer også endringer i hva studenter behøver å lære for å rustes for arbeidslivet. Som forfatter av bøker innen informatikk er dette noe jeg har vært nødt til å ta innover meg. Valg av tema og fremstilling av konsepter er nødt til å være generell nok til å holde seg relevant i et skiftende fagfelt, samtidig som relevansen her og nå, i konteksten av nåværende kunnskap, er viktig for å treffe studenter i dag.

I 2020 utga jeg boken Programmering i skolen (Haraldsrud mfl., 2020) sammen med to medforfattere ved Universitetet i Oslo. Boken ble til som et supplement for vår EVU-undervisning i programmering etter fagfornyelsen i grunn- og videregående skole, hvor programmering skulle inn som del av realfagene (Utdanningsdirektoratet, 2020). En slik omlegging av fagene er, for forfatteres del, heldigvis varig. De forrige læreplanene var utarbeidet i 2006, fjorten år før nye læreplaner. Vi kan vente at nye læreplaner også får lang varighet, og det betyr igjen at boken kan holde seg relevant i mange år, selv om teknologien

beveger seg videre. Selv om det kan tenkes at verktøyene som benyttes i skolen kan komme til å endre seg, vil de grunnleggende konseptene som læres bort i skolen være de samme så lenge læreplanen er gjeldende.

Da jeg skrev boken Objektorientert programmering med Python (Løvold, 2024), var jeg i større grad nødt til å ta innover meg at fagfeltet og universitetsfagene er mer dynamiske. I arbeidet med boken tok jeg et større blikk på hva som er likt, og hva som er endret de siste ti årene. Fokuset i boken er på grunnleggende programmering og konsepter innen programmeringsparadigmet objektorientert programmering. Da jeg selv lærte programmering var dette i et annet språk, Java. For å finne relevante tema for boken så jeg på de store linjene; hva ble undervist da jeg fulgte emner med tilsvarende bok i 2012? Hva er nytt i undervisningen jeg gir i tilsvarende emne i dag? Konseptene som gjennomgås i boken ble nøye utvalgt på bakgrunn av disse refleksjonene, med eksempler fra programmeringsspråket Python. Dette språket benyttes nå nesten universelt i undervisning i grunnleggende programmering norsk UH-sektor.

Jeg har lenge irritert meg over at norskspråklige grunnkurs ved norske universiteter, også kurs jeg selv har holdt, benytter utvalgte (og ofte små) deler av lange og dyre bøker på engelsk. Dette var hovedmotivasjonen min da jeg skrev Objektorientert programmering med Python. Boken er lagt opp etter egen undervisning, og dekker det jeg behøver. Målet var å kunne tilby en kortfattet lærebok innen objektorientert programmering som dekker de temaene som typisk undervises i objektorientert programmering ved norske universitet og høgskoler. Etter utgivelsen i 2024 er boken tatt inn som pensum ved UiT, UiO og UiS.

2 Utvikling av kurset INF-1049

I dette kapittelet vil jeg ta for meg kurset *INF-1049 – Introduksjon til beregningsorientert programmering*, som har vært mitt hovedarbeid når det kommer til undervisningsutvikling ved UiT.

2.1 Tradisjonelle forelesninger, høst 2021

Da jeg startet i stillingen min ved UiT i 2021, overtok jeg emnet *INF-1049 – Introduksjon til beregningsorientert programmering*. Dette kurset hadde tidligere gått som et tradisjonelt kurs med forelesninger og lab-øvelser der studentene fikk hjelp med arbeidskrav, som vist i tabell 1.

	Forelesning	Lab-øvelse
Timer per uke	2 - 4	2
Underviser	Foreleser	Studentassistent

Tabell 1: Ukentlig oppsett for kurset INF-1049 høsten 2021.

Det første året jeg hadde ansvar for kurset, gjennomførte jeg undervisningen etter denne modellen, med mulighet for både fysisk oppmøte og strømming/opptak

på grunn av pandemien. For å få løpende tilbakemeldinger fra studentene gjennom semesteret, benyttet jeg meg av et panel av studenter, slik som beskrevet av Pedersen (2021). Panelet ble satt sammen i første forelesning, og i utgangspunktet var det om lag fem studenter som meldte interesse for å være med. Etter noen få uker var det kun to studenter igjen i panelet, og vi hadde ukentlige møter der vi diskuterte undervisningen og undervisningsopplegget. Dette ga meg noen verdifulle tilbakemeldinger:

- Studentene liker forelesninger.
- Om lag halvparten av studentene møter opp til forelesninger.
- Studentene synes arbeidskravene er (for) vanskelige.
- Det er få studenter som møter opp til lab-øvelser.

For meg virket det paradoksalt at arbeidskravene, som tross alt bygger direkte på pensum som har vært forelest, oppleves som vanskelige for studentene. Dette var ikke store oppgaver, og for en som har forstått det pensum som er gjennomgått skulle det være en smal sak å løse disse. Videre syntes jeg også det var rart at det var så få som møtte opp til lab-øvelsene, som nettopp var til for at de skulle ha en arena der de kunne få hjelp av studenter med mer erfaring.

2.2 Et første forsøk med aktiv forelesning, høst 2022

Høsten 2022 var vi fri fra pandemi-restriksjoner, og vi kunne legge opp til at fysisk oppmøte var normen ved alle undervisningssituasjoner. Min medfaglærer Morten Grønnesby og jeg hadde erfaringene fra året før med oss, og ønsket å bevege oss mer i retning av aktiv undervisning. Vi beholdt lab-undervisningen med hjelpelærere og arbeidskravene fra året før, men i forelesningene forsøkte vi å få til mer diskusjon. Vi vekslet fortløpende på å gi oppgaver som studentene løste i auditoriet, og som vi så løste i fellesskap for å diskutere ulike måter å angripe problemer fra pensum på. Ikke bare var dette tenkt som en mulighet til samtale mellom faglærer og studenter, men også å gjøre det mer lukrativt å møte opp til forelesningene. Vi beholdt digital løsning med strømming og opptak, men mye ble kuttet derfra av personvernhensyn ettersom vi hadde samtaler med studentene i auditoriet.

Dette fungerte til en viss grad, men vi oppnådde ikke større oppmøte slik vi ønsket. Jeg vil hevde at dette skapte et større skille mellom "sterke" og "svake" studenter – de som møtte opp var stort sett de som allerede hadde mye kunnskap, og disse fikk nok mye ut av denne undervisningsformen. De svakere studentene, som vi gjerne skulle nådd, møtte opp i mindre grad og fikk også mindre ut av direktestrøm og opptak. Dette understøttes av karakterstatistikken som viste signifikant flere toppkarakterer og strykkarakterer i 2022, enn i 2021.

2.3 Vekselforelesninger, høst 2023

Høsten 2023 gjorde medfaglærer Morten Grønnesby og jeg endringer i oppsettet på kurset. Vi gikk vekk fra 2-4 timer forelesning i uken, og over til to timer forelesning tidlig i uken, etterfulgt av to timer seminar med problemløsning i auditorium i slutten av uken (som vist i Tabell 2). Vi beholdt lab-øvelser med studentassistenter som et tilbud ved siden av fellesundervisningen.

	Forelesning	Problemløsning	Lab-øvelse
Timer per uke	2	2	2
Underviser	Foreleser	Foreleser	Studentassistent

Tabell 2: Ukentlig oppsett for kurset INF-1049 høsten 2023.

Problemene vi valgte til problemløsningsseminarene, fulgte tett temaene som var forelest tidligere i uken. I et typisk seminar fikk studentene utdelt en oppgave som de så brukte om lag 60 minutter på å løse gruppevis. Gruppene ble satt sammen av studentene selv, og vi forsøkte etter beste evne å få studenter som satt alene til å gå sammen med eksisterende grupper. Underveis i seminaret ble det gitt hjelp til studenter som eventuelt lurte på noe. Den siste halvtimen av seminaret ble brukt til gjennomgang av problemet som ble gitt, med forelesers forslag til løsning. Siden problemene ble formulert relativt åpent, kunne dette lede til gode diskusjoner om ulik fremgangsmåte mellom underviser og studentene.

2.4 Problembasert læring, høst 2024

Etter erfaringer fra tidligere år, la vi i 2024 om kursets undervisning til rene problembaserte seminarer. På forhånd gjorde vi et stort arbeid med læringsmoduler i Canvas som inneholdt både tekst og video for hver ukes pensum, slik at studentene hadde noe å støtte seg på. I stedet for tradisjonelle forelesninger og lab-øvelser, hadde vi fire timer i uken med seminarer i større seminarrom, hvor studentene ble plassert i grupper for arbeid.

	Problembasert seminar
Timer per uke	4
Underviser	Faglærere og studentassistenter

Tabell 3: Ukentlig oppsett for kurset INF-1049 høsten 2024.

Å legge om til problembasert læring krevde mye, både av oss som faglærere og våre studentassistenter. Om man skal følge den problembaserte metodologien strengt, skal underviserne aldri gi svar på spørsmålene som stilles, men heller peke studentene i en slik retning at de selv kan finne svaret på det de lurer på. For oss som forelesere var det utfordrende å sitte på et klart svar, og likevel ikke kunne gi dette, men heller å måtte peke dem i en retning slik at de selv kunne finne ut av det. Spesielt frustrerende var det når studentene satt fullstendig fast

i noe såre enkelt, og man ikke kunne svare *"gjør det på denne måten!"*. Mange ganger tok jeg meg selv i å gå for langt i å foreslå konkrete løsninger, og selv om studentene nok var takknemlige for det, er det ikke sikkert at det ga like dyp læring som om de selv hadde funnet ut av det.

Vi var også nødt til å lære bort denne undervisningsformen til våre studentassistenter. Tidligere hadde disse undervist i lab-øvelser, hvor studentene typisk stilte spørsmål og fikk konkrete svar og løsninger. I ettertid ser jeg at vi nok burde fokusert mer på å formidle undervisningsformen tydeligere for studentassistentene, for det ble etter hvert klart at deres praksis når det kom til hvordan man svarer på spørsmål var meget varierende.

For studentene var det naturligvis også utfordrende å eksponeres for denne undervisningsmetoden. Ettersom INF-1049 typisk var ett av tre fag de fulgte dette semesteret, og undervisningen i øvrige fag var annerledes, var det vanskelig å omstille seg til vår måte å undervise på. Vi fikk relativt mange klager fra studenter etter at de i første seminar ble ønsket velkommen og fikk en oppgave utdelt, uten noen form for forelesning over temaet i forkant.

Vi så fort at oppmøtet til seminarene gikk ned, og i siste halvdel av semesteret gjorde vi noen justeringer i undervisningen. Problemer ble fortsatt gitt til gruppene, men i etterkant av problemløsningen ga vi en mini-forelesning på 10-15 minutter. Dette ble godt tatt imot av studentene, som rapporterte at de nå lærte mye mer ved oppmøte.

Som nevnt i kapittel 1.1, rapporterte studentene lavere tilfredshet med undervisningen dette året, sammenliknet med tidligere. Dette er nok ikke uventet, ettersom de støtte på mye mer frustrasjon gjennom semesteret, enn de ville gjort om de møtte opp i forelesninger. Denne frustrasjonen, og følelsen av at man hele tiden satt fast, førte nok dessverre til at en god del av studentene sluttet å møte opp. Likevel så vi at karaktersnittet gikk opp dette året. Om det skyldes undervisningsformen eller årskullet er det vanskelig å si noe om, men personlig tror jeg at de som møtte opp til undervisning i større grad ble i stand til å løse problemer på egen hånd, enn studenter fra tidligere år.

I etterkant ser jeg at dette forsøket var relativt radikalt, og at det kanskje ikke var så godt egnet for våre studenter som jeg hadde håpet på forhånd. Studentene som følger INF-1049 er typisk i starten av sitt første studieår, og det kan godt tenkes at studenter lengre ut i studieløpet i større grad ville sett nytten av undervisningsformen. I 2025 vil vi fortsette med problembasert læring i første semester, men kun i andre halvdel av semesteret. Dette beskrives nærmere i kapittel 2.5.

2.5 Porteføljearbeid og nye emner, høst 2025

Til og med høsten 2024 tilbød Institutt for informatikk to kurs på ti studiepoeng i grunnleggende programmering. Ett av disse var tidligere nevnte INF-1049, som typisk ble fulgt av alle programstudenter ved NT-fakultetet (med noen

unntak), og det andre var INF-1100 som ble fulgt av programstudenter på informatikk. Selv om de benyttede programmeringsspråkene i kursene er ulike, var mange av de underliggende konseptene de samme, spesielt i første halvdel av semesteret.

For å utnytte undervisningsressursene ved instituttet ble det derfor besluttet at en ny kursstruktur skal på plass fra og med høsten 2025. Den nye strukturen innebærer at alle studenter følger et felles kurs på fem studiepoeng (INF-0101) første halvdel av semesteret, mens studentene som tidligere ville tatt INF-1049 skal over i ett kurs (INF-0102) og informatikk-studentene skal over i et annet kurs (INF-0103) siste halvdel av semesteret. Dette vil altså bli en Y-vei, der studentene først deles i semesterets andre halvdel.

Å utarbeide nye kurs er et stort arbeid som krever planlegging av både pensum og undervisningsform, og dette er et pågående arbeid som jeg tar del i. I 2024 og 2025 har vi hatt undervisningsseminar ved instituttet, og vi har nedsatt en arbeidsgruppe bestående av tidligere forelesere fra begge tidligere kurs. Gjennom en systematisk gjennomgang av pensum i begge de tidligere kursene, har vi identifisert hvilke konsepter de har til felles, og hva som er fagspesifikt. Dette ledet oss til nevnte Y-vei, hvor de felles konseptene foreleses i første halvdel av semesteret for samtlige studenter.

Vi har støtt på flere utfordringer med utformingen av undervisning for det første felleskurset. For det første vil vi være nødt til å gjøre kursene mer programmeringsspråk-agnostiske, altså å frikoble konseptene mer fra eksemplene. Etter erfaringer med problembasert undervisning fra INF-1049 har vi ikke gått videre med problembasert læring i første halvdel av semesteret, men vil kjøre en mer tradisjonell forelesningsform med lab-øvelser ved siden av. Som et supplement til dette vil separate moduler for hver ukes pensum utarbeides i Canvas, hvor studentene kan se eksempler i video og tekst fra det programmeringsspråket de skal benytte i siste halvdel av semesteret. Dette tillater oss å være mer fokusert på konsepter i forelesningene, heller enn å gå inn på praktiske eksempler fra programmeringsspråk. Også arbeidskravene vil gis i to ulike varianter, hvor studentene må bestå én av variantene for å kunne gå opp til eksamen.

3 Programmering i skolen

Gjennom min tid som universitetslektor, først ved Universitetet i Oslo og senere ved UiT, har ett av mine viktigste fokusområder vært innføringen av programmering i skolen. Fra og med høsten 2020 ble det innført nye læreplaner i grunn- og videregående skole, og programmering ble tatt inn som en del av pensum i realfagene (Utdanningsdirektoratet, 2020).

Frem til midten av 2000-tallet, og ved noen universiteter og høgskoler enda senere, var kurs i programmering ikke del av lektorutdanningen i realfag. Dette betød at lektorer utdannet før dette i all hovedsak manglet kompetanse på programmering. At programmering skulle inn i klasserommet og undervises av

skolens lærere ble derfor en stor utfordring. Støtte fra universitetssektoren til etter- og videreutdanning av lærere ble meget viktig i årene frem mot innføringen av fagfornyelsen, og også i årene etter, for å ruste lærerne til å undervise etter de nye læreplanene.

3.1 ProFag – Programmering for fagene

Da jeg i 2018 jobbet ved Universitetet i Oslo, gikk jeg inn i en gruppe ved CCSE – Center for Computing in Science Education, som på den tiden var et senter for fremragende utdanning ledet av Anders Malthe-Sørenssen. I gruppen bestående av Knut Mørken, Cathrine Tellefsen, Andreas Haraldsrud, Henrik Sveinsson og undertegnede, utarbeidet vi en kursrekke for lærere i den videregående skolen. Denne kursrekken, som fikk navnet ProFag – Programmering for fagene, skulle bli et etterutdanningskurs for lærere i den videregående skolen. Senere ble kurset også utvidet til en egen kursrekke for lærere i ungdomsskolen.

Kurset ble pilotert høsten 2018, med lærere fra utvalgte skoler i Oslo og omegn. Vi valgte å legge opp kurset i fire halvdagssamlinger i klasserom, med hjemmearbeid mellom hver samling. På samlingene deltok 20-30 lærere som ble inndelt i grupper. Vi vekslet fortløpende gjennom kursdagene med korte forelesninger og gruppearbeid med oppgaver relatert til temaene som nettopp var forelest. Vår erfaring og tilbakemeldinger fra kursdeltakerne indikerte at dette var en god måte å undervise på. Korte forelesningsøkter etterfulgt av oppgaveløsning i grupper er en vanlig undervisningsform i skolen, og noe lærerne var godt kjent med fra eget arbeid. Ved ikke bare å høre, men også å gjøre, fikk lærerne utfordret seg selv, og også kjent på følelsen av å sitte fast, slik som deres elever gjør i klasserommet.

Etter piloteringen i 2018 søkte vi støtte fra Utdanningsdirektoratet for finansiering av videre kursing. Vi mottok støtte gjennom Desentralisert ordning for kompetanseutvikling i skolen (DEKOMP), og skalerte opp med flere parallelle kursrekker for lærere i fylkeskommunene Oslo og tidligere Viken. Arbeidet ledet også til boken Programmering i skolen (Haraldsrud mfl., 2020), som jeg forfattet sammen med Andreas Haraldsrud og Henrik Sveinsson.

I 2021 ble jeg tilsatt som universitetslektor ved UiT og gikk ut av arbeidet med ProFag. I ettertid er kurset videre utviklet, og tilbys fortsatt som del av EVU-tilbudet i realfag ved Universitetet i Oslo.

3.2 Studier av programmering i skolen

Innføringen av programmering i realfagene i norsk skole er et stort eksperiment, og har ved seg mange ubesvarte forskningsspørsmål. Lærer egentlig elevene matematikk bedre ved å benytte programmering som verktøy? Hva mener lærerne om innføringen av programmering? For oss ved universitetet er det også relevant å forstå hvilken kompetanse innen programmering kommende generasjoner av studenter besitter, slik at vi kan tilpasse undervisningen til målgruppen.

Jeg har veiledet tre studenter til mastergrad innen disse temaene. Høsten 2020 og våren 2021 var innføringen av nye læreplaner fersk, og min student Erlend Stenlund og jeg ønsket å se på hvilke holdninger lærerne hadde til innføringen. Vi fant at lærerne i hovedsak var positive til innføringen, men at kompetansehevingstiltak ved skolene lokalt og i UH-sektoren sentralt var viktig for å få lærerne opp på et slikt nivå at de kunne stå i klasserommene og undervise etter nytt pensum (Stenlund, 2021).

Etter at de første kullene med elever i videregående skole hadde vært igjennom det nye pensumet i matematikk med programmering noen år, ble det aktuelt å se på hvordan programmeringen egentlig hadde påvirket deres dybdelæring i matematikkfaget. Målet med innføringen av programmering var av UDIR definert som en støtte til innlæring av matematikkonsepter, og min student Ane Sæther og jeg ønsket å finne ut av hvorvidt programmering faktisk hjalp med dette. Våre funn indikerte at elevene ikke nødvendigvis lærte seg matematikken bedre gjennom å benytte programmering i undervisningen (Sæther, 2022). Dette er et interessant funn som potensielt kan brukes som innspill til senere høringer om endringer i læreplanverket.

Som tidligere nevnt får innføringen av programmering i skolen også konsekvenser for undervisningen vi gjør ved universitetet. Undervisningen vi gir i første semester tar fortsatt utgangspunkt i at studentene ikke har noen kjennskap til programmering på forhånd, og begynner derfor fra det helt grunnleggende. Hvordan studenter som tidligere har hatt programmering i skolen opplever undervisningen, og hvordan deres tidligere programmeringserfaring påvirker deres forståelse og mentale modeller av programmering når de starter som studenter innen informatikk, er et interessant tema. For å undersøke dette gjorde min student Thomas Eide og jeg en komparativ studie med undersøkelser av forståelse hos førsteårsstudenter som hadde, og ikke hadde, hatt programmering i skolen. Våre funn indikerer at studentenes kompetanse i hovedsak er bundet til språket de har benyttet i skolen (typisk Python) og dets syntaks, heller enn de bakenforliggende konseptene som utgjør deler av et program (Eide, 2024). Som underviser i programmering ved universitetet er dette noe jeg siden har tatt høyde for; studentene kjenner kanskje språket og syntaksen, men semantikken i programmene de skriver er de ikke nødvendigvis kjent med.

4 Plan for videre utvikling

Jeg har mye erfaring fra undervisning av studenter på ulike nivå, både i ordinære kurs og som del av EVU-studier. Likevel er det mye jeg ønsker å se nærmere på i tiden fremover. Et viktig punkt der jeg selv føler at jeg mangler erfaring og kunnskap er kollegaveiledning.

Når man gjør seg erfaringer med undervisning og undervisningsformer, enten det har fungert godt eller mindre godt, er det nyttig for institusjonen som helhet å få innblikk i dette. Ved Institutt for informatikk har vi gode forum for formidling

av egen forskning, mens undervisning lett havner litt i bakgrunnen. En felle det kan være viktig å unngå, er at noen ved institusjonen har gjort forsøk med undervisning uten å dele erfaringene videre med kollegiet. Da risikerer man at andre bruker mye tid på å gå opp en vei i egen undervisning som allerede er utprøvd av andre, uten at man kan høste av deres erfaring.

Jeg har som mål fremover å løfte undervisning som en like viktig del av universitetets samfunnsoppdrag som forskning. For å få til dette trengs flere arenaer for erfaringsdeling. Institutt for informatikk har for tiden ett seminar i semesteret viet til undervisning. På de hyppige instituttmøtene for vitenskapelig ansatte er gjerne "oppdatering fra forskningsgruppene" et sentralt punkt, mens undervisning sjelden nevnes.

For å fremme instituttets undervisning har jeg et mål om å sette ned en "undervisningsgruppe" på tvers av forskningsgruppene. En slik gruppe kan være en ressurs for alle som underviser kurs hos oss, og jeg har et håp om at det vil være en naturlig plass å oppsøke for kolleger som ønsker å utvikle egen undervisning. Til tross for at undervisning er bundet til eget fag, er det mer generelle linjer som kan trekkes, typisk i form av erfaring med undervisnings*former*. Selv er jeg nysgjerrig på hva andre har gjort i sin undervisning, hva som har fungert, hva som ikke har fungert, og ikke minst hvorfor.

Ved å observere andres undervisning kan man i større grad få øynene opp for studentenes syn, enn man får når man står i underviserrollen selv. I kurs i universitetspedagogikk går man typisk sammen i grupper og observerer hverandres undervisning på tvers av fagfelt, med mål om å gi tilbakemeldinger i etterkant. Selv kunne jeg godt tenke meg i større grad å ta del i andres undervisning som observatør, og for meg hadde det vært positivt å ha andre til stede under min undervisning for å få tilbakemelding på mitt arbeid. En undervisningsgruppe ved instituttet kunne koordinert slike observasjoner på tvers av fagfelt og stimulert til tilbakemeldinger og erfaringsdeling.

Referanser

Eide, T. V. (2024). *Integration of programming in Norwegian schools: The effects of prior programming experience on students in a university-level programming course* [Masteroppgave, UiT Norges arktiske universitet].

Haraldsrud, A. D., Sveinsson, H. A., & Løvold, H. H. (2020). *Programmering i skolen*. Universitetsforlaget.

Krathwohl, D. R. (2002). A revision of Bloom's taxonomy: An overview. *Theory into practice*, *41*(4), 212–218.

Løvold, H. H. (2024). *Objektorientert programmering med Python*. Fagbokforlaget.

Løvold, H. H., Lindsjørn, Y., & Stray, V. (2020). Forming and assessing student teams in software engineering courses. *Agile Processes in Software Engineering and Extreme Programming–Workshops: XP 2020 Workshops,*

Copenhagen, Denmark, June 8–12, 2020, Revised Selected Papers 21, 298–306.

Mazur, E., & Somers, M. D. (1999). *Peer instruction: A user's manual*. American Association of Physics Teachers.

Pedersen, E. (2021). Experiences with informal student panel for continuous formative course evaluation. *Nordic Journal of STEM Education*, *5*(1).

Rørnes, K. M., Runde, R. K., & Jensen, S. M. (2019). Students' mental models of references in Python. *Norsk IKT-konferanse for forskning og utdanning*.

Stenlund, E. (2021). *Programmering og fagfornyelsen* [Masteroppgave, Universitetet i Oslo].

Sæther, A. (2022). *Can programming improve mathematical conceptual understanding? A qualitative study about the implementation of programming in the subject mathematics* [Masteroppgave, Universitetet i Oslo].

Utdanningsdirektoratet. (2020). Fagfornyelsen [https://www.udir.no/laring-og-trivsel/lareplanverket/fagfornyelsen/ Hentet 6. juni 2025].

Wood, D. F. (2003). Problem based learning. *Bmj*, *326*(7384), 328–330.

Zingaro, D., & Porter, L. (2015). Tracking student learning from class to exam using isomorphic questions. *Proceedings of the 46th acm technical symposium on computer science education*, 356–361.

Vedlegg 1

I dette vedlegget vil jeg legge frem min erfaring som underviser og emneansvarlig for kurs på universitetsnivå. Jeg vil også legge frem min erfaring som veileder og hvilke resultater som er oppnådd av studenter jeg har veiledet. Til slutt finnes en oversikt over mitt arbeid som sensor.

Undervisningserfaring fra universitetskurs

Undervisningen jeg har gjort er sortert per år.

2017

- Enkeltforelesninger i kurset IN1000 – Objektorientert programmering ved UiO

2018

- Utvikling av- og pilotering av EVU-kurset ProFag (Programmering i Fagene) ved UiO
- Enkeltforelesninger i kurset IN1000 – Objektorientert programmering ved UiO

2019

- Medansvarlig for kurset IN2000 – Software Engineering med prosjektarbeid ved UiO
- Undervisning av EVU-kurset ProFag (Programmering i Fagene) ved UiO
- Enkeltforelesninger i kurset IN1000 – Objektorientert programmering ved UiO

2020

- Medansvarlig for kurset IN2000 – Software Engineering med prosjektarbeid ved UiO
- Undervisning av EVU-kurset ProFag (Programmering i Fagene) ved UiO
- Enkeltforelesninger i kurset IN1000 – Objektorientert programmering ved UiO

2021

- Emneansvarlig for kurset INF-1049 – Introduksjon til beregningsorientert programmering ved UiT
- Undervisning i kurset INF-1400 – Objektorientert programmering ved UiT

- Utarbeiding og kursansvar for Dekomp-kurs i realfaglig programmering (se vedlegg 2)
- Emneansvarlig for kurset IN1000 – Objektorientert programmering ved UiO

2022

- Emneansvarlig for kurset INF-1049 – Introduksjon til beregningsorientert programmering ved UiT
- Emneansvarlig for kurset INF-1400 – Objektorientert programmering ved UiT
- Kursansvar for Dekomp-kurs i realfaglig programmering (se vedlegg 2)
- Emneansvarlig for kurset IN1000 – Objektorientert programmering ved UiO
- Enkeltforelesninger i kurset DIGHEL4360 – Informatikk for helsearbeidere ved UiO

2023

- Emneansvarlig for kurset INF-1049 – Introduksjon til beregningsorientert programmering ved UiT
- Emneansvarlig for kurset INF-1400 – Objektorientert programmering ved UiT
- Kursansvar for Dekomp-kurs i realfaglig programmering (se vedlegg 2)
- Emneansvarlig for kurset IN1000 – Objektorientert programmering ved UiO
- Enkeltforelesninger i kurset DIGHEL4360 – Informatikk for helsearbeidere ved UiO

2024

- Emneansvarlig for kurset INF-1049 – Introduksjon til beregningsorientert programmering ved UiT
- Emneansvarlig kurset INF-1400 – Objektorientert programmering ved UiT
- Emneansvarlig for kurset IN1000 – Objektorientert programmering ved UiO
- Enkeltforelesninger i kurset DIGHEL4360 – Informatikk for helsearbeidere ved UiO

2025

- Arbeidsgruppe for utarbeiding av nye kurs, jf. porteføljearbeid ved IFI, UiT
- Emneansvarlig kurset INF-1400 – Objektorientert programmering ved UiT
- Emneansvarlig for kurset IN1000 – Objektorientert programmering ved UiO

Erfaring som veileder

- Masteroppgave Erlend Stenlund, *Programmering og fagfornyelsen*
- Masteroppgave Ane Sæther, *Can programming improve mathematical conceptual understanding? A qualitative study about the implementation of programming in the subject mathematics*
- Masteroppgave Thomas Vatne Eide, *Integration of programming in Norwegian schools: The effects of prior programming experience on students in a university-level programming course*
- Masteroppgave Obrestad Schei, *Hvordan introdusere versjonskontroll til førsteårsstudenter? Implementasjon og evaluering av en interaktiv nettside for å lære om Git*
- Masteroppgave Elias Gåre Opstad, *Hvordan introdusere DevOps til studenter?*
- Masteroppgave Mohammad Zahirul Islam, *Comfortable remote presence in VR using video streaming with motion compensation*

Sensur

Oversikt over sensurarbeid, internt og eksternt. Inkluderer sensur av kontinuasjonseksamen og klagesensur.

- UiT – Norges arktiske universitet
  - INF-1400 – Objektorientert programmering (2021, 2022, 2023, 2024, 2025)
  - INF-1049 – Introduksjon til beregningsorientert programmering (2021, 2022, 2023, 2024)
  - INF-1101 - Datastrukturer og algoritmer (2022, 2023, 2024)
  - INF-1100 – Innføring i programmering og datamaskiners virkemåte (2022, 2023)
  - INF-3983 – Capstone Project in Computer Science (2023)
  - Masteroppgaver i informatikk (2021, 2022, 2023)

- Universitetet i Oslo
  - INF1000 – Grunnkurs i objektorientert programmering (2017, 2018)
  - INF1001 – Grunnkurs i objektorientert programmering, pilotemne (2017, 2018)
  - IN1000 – Introduksjon til objektorientert programmering (2019, 2020, 2021, 2022, 2023, 2024, 2025)
  - INF1010 – Objektorientert programmering (2017, 2018)
  - IN1010 – Objektorientert programmering (2023)
  - IN2000 – Software Engineering med prosjektarbeid (2019, 2021)
  - Masteroppgaver i informatikk (2019, 2020, 2021, 2022, 2023, 2024, 2025)
- NTNU: Norges teknisk-naturvitenskapelige universitet
  - TDT4100 – Objektorientert programmering (2019)
  - INFT2900 – Bacheloroppgave i informasjonsbehandling (2025)
- Nord universitet
  - IKT1004 – Internett, sikkerhet og tjenester (2023, 2024)
- OsloMet
  - MAIKT-OP – Programmering i skolen (2019)
  - PRG6100 – Innføring i programmering for lærere for trinn 5-10 (2019)

Vedlegg 2

Undervisning ved Skolelaboratoriet

Skolelaboratoriet for realfag og teknologi er en tverrfaglig arena ved NT-fakultetet på UiT. Laboratoriet tilbyr skolebesøk, og er en viktig del av fakultetets utadrettede aktivitet. I dette vedlegget legger jeg frem aktiviteter som jeg har deltatt i og undervist ved laboratoriet.

Lærerkurs

Halv- og heldagskurs for lærere i realfaglig programmering etter ny læreplan K20.

- 27. februar 2020, kurs holdt ved Tromsdalen vgs
- 27. april 2020, online på grunn av corona
- 15. oktober 2020, online på grunn av corona

Dekomp-kurs

Kurs i realfaglig programmering for Troms og Finnmark fylkeskommune. Del av kursrekker finansiert på Dekomp-midler. Hver kursrekke inneholdt fire halvdagskurs online, med egenarbeid for deltakere mellom hvert kurs. Her var jeg hovedansvarlig fra Institutt for informatikk, og underviste sammen med ansatte ved ILP og Vitensenteret.

- Første kursrekke
  - 14. september 2021, samling 1
  - 24. november 2021, samling 2
  - 9. februar 2022, samling 3
  - 21. mai 2022, samling 4
- Andre kursrekke
  - 3. mars 2022, samling 1
  - 5. mai 2022, samling 2
  - 14. oktober 2022, samling 3
  - 4. november 2022, samling 4
- Tredje kursrekke
  - 11. oktober 2022, samling 1
  - 8. november 2022, samling 2
  - 26. november 2023, samling 3

  - 20. april 2023, samling 4

Forskningsdagene

Rekrutteringsarrangement ved UiT for elever i videregående skole før valg av studier. Aktiviteter på vegne av Institutt for informatikk.

- 16. til 27. september 2020
- 21. september til 2. oktober 2022

Air:bit

Større prosjekt om luftkvalitet ved Institutt for informatikk, der elever og lærere kommer til universitetet for å bygge og programmere støvsensorer. Elevene gjennomfører i løpet av et halvt år et forskningsprosjekt og presenterer. Lærerne kommer til universitetet før gjennomføringen av prosjektet og deltar på et lærerkurs.

- Lærerkurs
  - 12. november 2020
  - 27. april 2022
  - 11. og 12. november 2024
- Samling for elever
  - 17. februar 2021, Kroken skole
  - 21. mars 2021, Kroken skole
  - 22. april 2022, Kongsbakken vgs
  - 3. mai 2022, Kongsbakken vgs
  - 7. februar 2024, Alta vgs
  - 5. februar 2025, Meløy vgs

Scratch-programmering

Programmering i det visuelle språket Scratch. Elever fra barne- ungdoms- og videregående skole kommer til campus, eller ved utreise til skolen.

- 16. mars 2022, kurs for Tromstun skole
- 18. mai 2022, kurs for Slettaelva skole

Bestill en programmerer

Klasseromsundervisning for elever i ungdoms- og videregående skole, enten på campus eller ved reise til skolen. Kursene varer typisk 2-3 timer og handler om programmering direkte knyttet opp mot fagene matematikk, naturfag, fysikk og teknologi- og forskningslære.

- 21. august 2020, Tromsdalen vgs
- 26. august 2020, Bardufoss vgs
- 4. september 2020, Husøy og Botnhamn skoler
- 17. september 2020, Tromsdalen vgs
- 18. september 2020, tre ulike kurs (hel dag) for Tromsdalen vgs
- 29. september 2020, Tromsdalen vgs
- 30. september 2020, Tromsdalen vgs
- 14. oktober 2020, fire ulike kurs (hel dag) for Alta vgs
- 21. september 2021, Breivang vgs
- 2. november 2021, Tromsdalen vgs
- 11. februar 2022, Trondjord skole
- 9. september 2022, Tromsdalen vgs
- 23. september 2022, Breivang vgs
- 27. januar 2023, Tromsdalen vgs

Kreativ programmering

Oppfølger til kurset ≪Bestill en programmerer≫ med nytt pensum, både for ungdoms- og videregående skole. Kurset er mer knyttet til informatikk som fagfelt, og mindre direkte til skolefagene. Nytt av 2025.

- 28. mai 2025, kurs for Hansnes skole

App-utvikling

Kurs der elever i ungdomsskolen utvikler en app til mobiltelefon.

- 4. april 2022, kurs for Grønnåsen skole
- 9. februar 2023, kurs for Bardufoss ungdomsskole

Lær å lage dataspill

Et mini-kurs for elever i videregående skole der de arbeider med å lage et lite dataspill.

- 15. juni 2021, kurs for Nordkjosbotn vgs

Arduino – mikrokontrollere og C-programmering

Kurs for elever som tar informasjonsteknologi i videregående skole. I dette kurset lærer elevene å benytte og programmere mikrokontrollere som de bruker i et lite prosjekt.

- 3. juni 2022, kurs for Bardufoss vgs