

Meistringslæring i intro programmering: plan for eit nytt emne

G. Hansen, G. S. Korpås, A. Kirknes, J. L. Magnussen, J. X. Skøien og G. Sindre, *NTNU*

SAMANDRAG: Hausten 2023 startar eit nytt programmeringsemne spesielt for studentar på lektorutdanning i realfag ved NTNU. Opplegget vil vere basert på meistringslæring. I staden for avsluttande eksamen finst ein serie modultestar som studentane kan ta med individuell framdrift gjennom semesteret. I tillegg skal kvar student utføre eit individuelt prosjekt der programmering blir brukt med tanke på undervisning av realfag i skulen. Studentar må passere fleire testar og levere gradvis meir ambisiøst prosjekt jo betre karakter dei siktar mot. Artikkelen presenterer motivasjonen for korfor emnet blir lagt opp på denne måten, med gjennomgang av generell litteratur om meistringslæring, tilbakemelding og studieåtfærd, og meir spesifikk litteratur om andre sine erfaringar med meistringslæring i innleiande programmeringsemne, både i Noreg og utlandet. Deretter presenterer vi det planlagte undervisningsopplegget for emnet, og korleis studentar frå det aktuelle programmet har vore involvert i utviklingsprosessen. Til sist diskuterer vi moglege risiko og utfordringar som vil måtte handterast i opplegget, kva følgjeforskning som er tenkt utført undervegs, og korleis ein ser for seg vegen vidare.

1 INNLEIING: MOTIVASJON FOR DET NYE EMNET

1.1 Behov for tiltak i LUR-studieprogrammet

Våren 2022 blei det lyst ut prosjektmidlar frå Forvaltningsutvalet for Lektorutdanningane (FUL) ved NTNU med tanke på å betre desse utdanningane. Fleire prosjektsøknader vart innvilga, og ein av desse er den vi skal fortelje om i denne artikkelen – nemleg å lage ein eigen variant av IT Grunnkurs for studentar som går på det 5-årige masterprogrammet Lektorutdanning i Realfag (vidare i artikkelen kalla LUR). Eit slikt emne kan løyse fleire utfordringar for LUR-programmet på éin gong: (i) gje eit emne med læringsutbytte som er meir tilpassa akkurat LUR, ved å ta inn undervisningsaspektet ved IT, (ii) vere eit fagleg-sosialt lim i LUR-programmet, sidan dette programmet elles ikkje har hatt noko emne som er sams for, og berre for, studentane i første semester av LUR.

Årsaken til det siste problemet er at LUR-programmet har fem ulike retningar: matematikk-informatikk, matematikk-fysikk, matematikk-kjemi, matematikk-biologi og biologi-kjemi. Desse har dels ulike emnekombinasjonar, t.d. har biologi-kjemi-retningen mindre avansert matematikk enn dei andre. Alle retningane har hatt IT Grunnkurs i første semester, men matematikk-informatikk har da TDT4109 saman med studentar m.a. frå Data og Informatikk, medan dei andre retningane har TDT4110 saman med diverse siv.ing. og realfagsstudentar – og uansett er dette ein kjempestor klasse med meir enn tusen studentar, der det er vanskeleg å skape noko klassemiljø spesifikt for LUR.

1.2 Korfor meistringslæring i intro programmering?

Meistringslæring i seg sjølv er ikkje spesielt nytt eller innovativt. Kjende metodar innan meistringslæring, som *Learning for Mastery* (Bloom, 1968) og *Personalized System of Instruction* (PSI) (Keller, 1968) blei foreslått for meir enn 50 år sidan. Sentrale idear i meistringslæring er modularisering og individuell framdrift («self pacing»). Pensum blir delt opp i mange modular slik at kvar einskild modul blir liten nok til å ikkje vere overveldande, og studenten må dokumentere meistring i éin modul for å gå vidare til neste modul. Kvar student kan velje sitt eige tempo gjennom serien av modular, og det blir lagt stor vekt på studenten sin kommunikasjon med læringsassistent (kalla «tutors» av Keller) undervegs i semesteret. Det var stor internasjonal interesse rundt denne typen opplegg på 70- og 80-talet og mange empiriske studier indikerte betre læringsutbytte enn for vanlegare opplegg med forelesingar pluss avsluttande eksamen (Kulik, Kulik, & Cohen, 1979). Likevel avtok interessa for meistringslæring på 90-talet, av ulike årsaker. Heidi Eyre (2007) spurde: “*Keller's Personalized System of Instruction: Was it a fleeting fancy or is there a revival on the horizon?*” Eitt av argumenta for gjenoppliving var at digitalisering av høgare utdanning kunne mogleggjere større grad av automatisering av meistringstestar og relatert tilbakemelding – og dermed redusere arbeidsbyrda som slike undervisningsopplegg la på fagstaben. Sidan programmering er eit fag som kan egne seg særleg godt for automatisk evaluering og tilbakemelding, kan nettopp slike emne vere ein viktig del av ein horisont der meistringslæring igjen

blir meir populært (Sindre, 2018). Det tenkte opplegget for emnet IT1001 Informasjonsteknologi grunnkurs for LUR-studentane er eitt forsøk på å få meistringslæring til å fungere i ein moderne kontekst med stor grad av autoregning av testane undervegs.

2 RELEVANT LITTERATUR

I denne seksjonen vil vi presentere relevant bakgrunns litteratur på to område: (i) meistringslæring både generelt og spesifikt om erfaringar i intro programmeringsfag, og (ii) forskningslitteratur om tilbakemelding og studieåttferd som kan være relevant for det tenkte emnedesignet.

2.1 Mestringslæring generelt, og i intro programmering

Som nemnt i Introduksjon, blei fleire metodar for meistringslæring foreslått for meir enn 50 år sidan, og empiriske studiar har i snitt vist ein positiv læringseffekt for denne typen intervensjon (Kulik et al., 1979). Utan meistringslæring er emne på universitetsnivå typisk forma slik at alle studentar skal følgje same framdrift (t.d. i form av ein serie førelesingar og obligatoriske øvingar veke for veke) og bruke like lang kalendertid (den tida semesteret varer, fram til eksamen). Læringsutbyttet kan derimot vere svært variabelt, alt frå høg grad av meistring til nesten ingenting (t.d. ved strykkarakter). Ideen bak meistringslæring er derimot at alle studentar skal oppnå meistring, men da må kunne bruke ulik tid og ha ulik framdrift gjennom emnet. For å oppnå dette blir emnet delt opp i modular, der studenten må ha meistra den første modulen for å gå vidare til den andre, og så bortetter.

Det har vore fleire forsøk på bruk av meistringslæring i intro programmeringsfag. Ein oversiktsstudie (Garner, Denny, & Luxton-Reilly, 2019) viser fleire døme på vellukka bruk, men konkluderer likevel med at litteraturen er avgrensa og det er vanskeleg å trekke overgripande konklusjonar. Det er også monalege variasjonar i opplegget. Somme har framleis ein eksamen ved semesterslutt som er det som gir karakter (Treviño & Cavazos, 2018) eller iallfall deler av karakteren. Til dømes hadde (Campbell, Petersen, & Smith, 2019) ved Univ. Toronto (Canada) sitt opplegg ein avsluttande eksamen som talde 45%, medan meistringstestar (x7) og øvingar (x3) utgjorde resten av karakteren. Andre har opplegg som kan kallast «contract grading», der studentane kan velje ambisjonsnivå på førehand, t.d. (LeJeune, 2010). Dette elementet finst også i opplegget rapportert frå Universitetet i Agder, som har brukt PSI i eitt av sine programmeringsfag (Purao, Sein, Nilsen, & Larsen, 2017). Resultata frå Agder er lovande, men som så mange andre forsøk med meistringslæring var det problem med prokrastinering hos ein del studentar, slik at ulike supplerande tiltak blei prøvd ut frå år til år. Utvikling over tid har også vore eit sentralt trekk i opplegget ved Univ. Otago (New Zealand). Det første forsøket deira hadde avsluttande eksamen etter at studentane hadde hatt eit opplegg med individuell framdrift (McCane, Ott, Meek, & Robins, 2017) men i seinare iterasjonar gjekk dei bort frå avsluttande eksamen men prøvde andre tiltak for å motverke prokrastinering (Ott, McCane, & Meek, 2021).

Ved FUCG (Brasil) har likeeins meistringslæring vore brukt i ei årrekke, med tidlege erfaringar rapportert i artikkelen (de Macêdo Morais, Figueiredo, & Guerrero, 2014) og ei vidareutvikling med spillifisering av meistringsstigen i (de Pontes, Guerrero, & de Figueiredo, 2019). Dei opererte med 10 modular. Opplegget til (Alvarez, Samary, & Wise, 2023) hadde derimot berre 4 modular, som dermed var monaleg større (kvar meint å vare i tre veker). I artikkelen (Stegeman, 2019) blir det fortalt om ein teststige for meistringslæring spesifikt for kodeforståing, som var blitt brukt i tre ulike programmeringskurs ved Univ. Amsterdam. Her var testane obligatoriske, men talde ikkje på karakteren. Izu et al. (2023) presenterer ikkje eit spesifikt undervisningsopplegg, men samanliknar meistringslæring med læring frå feil («productive failure»), dvs. ta ein test før ein har lært materialet, for å sjå kor lite ein får til, og dermed også sjå at ein blir betre når ein deretter set seg inn i stoffet. Produktive feil kan også vere eit element i vårt opplegg, da studentane kan prøve ut formativt dei same testane som deretter blir gjort summativt, og dermed lære av feil som dei gjer. Jayazeri (2015) fortel om eit opplegg som kombinerer ein meistringsstige med eit prosjektarbeid, ein kombinasjon som vi også har. Det er likevel ein viktig skilnad: Jayazeri sitt opplegg hadde meistringsstigen med automatiske testar først, som ein inngangsport til prosjektarbeidet som blei utført i andre halvdel av semesteret, medan vårt opplegg har prosjekt og testar i parallell.

2.2 Tilbakemelding som sosial prosess: dialog, berekraft og sjølvregulering

I dette prosjektet er meistring, framdrift og tilbakemelding gjennom dialog mellom partane viktig. Derfor samsvarer dette prosjektet med fokus på meistringslæring godt med det nye paradigmet innanfor forskningslitteraturen på tilbakemelding i høgare utdanning i dag. Før har tilbakemelding blitt sett på som

noko som blir gitt til studentane, men dette endra seg rundt 2010, og tilbakemelding blir no sett som ein prosess der studentane er aktive deltakarar (Dawson et al., 2019). For å gjere det mogeleg for studentane å ta en aktiv del i ein tilbakemeldingsprosess er *dialog* framheva som eit sentralt verkemiddel (Winstone & Carless, 2019). Her kan det både vere snakk om dialog mellom student og lærar, studentane seg imellom og indre dialog gjennom sjølvvurdering. Alle desse vil stå sentralt i dette prosjektet. Dialog blir sett på som essensielt for å oppnå det vi gjerne kallar en meir berekraftig tilbakemeldingspraksis der målet er at studentane skal bli sjølvregulerte ved å setje seg eigne mål, velje effektive læringsstrategiar og kontinuerleg utvikle disse (Johnson & Molloy, 2018). For studentane i det nye emnet prosjekt vil det å utvikle eigen framdriftsplan, monitorere eiga framdrift, evt. endre plan, metodar og innsats, samt ein kontinuerleg og nær kommunikasjon med medstudentar og læringsassistentar, vere sentrale element på veg mot utvikling av meir sjølvregulerande evner.

Som nemnt ovanfor er eit viktig mål i meistringslæring at alle studentar skal oppleve meistring. For dette prosjektet vil det derfor vere viktig å utvikle ein tilbakemeldingsprosess for studentane som fokuserer på læring, og det å lære av eigne mistydingar. Når en tilbakemeldingsprosess handlar om kva den einskilde kan og bør gjere før neste vurderingssituasjon, så opnar det opp for at student kan utvikle seg. Dette kan bidra til læring, både direkte gjennom vidare innsats og indirekte ved at det motiverer studentane til å sjå at innsats nyttar (Dweck & Master, 2012). Eit viktig element her er å skape ein meistringskultur der det er lov å misforstå og gjere feil, og der studentane kan lære av feila (Hattie & Clarke, 2018). Utan en klasseromsfilosofi som ser på feil som eit høve for læring, og som oppmuntrar til ærleg refleksjon omkring dette, vil læringsføremonane ved ein meir dialogisk tilbakemeldingspraksis sannsynlegvis bli kompromittert (Molloy, Noble, & Ajjawi, 2019).

3 PLAN FOR EMNET

3.1 Modulinndeling

For den første gjennomføringa av IT1001 tenkjer vi å leggje opp meistringsstigen med 9 modular. Desse modulane gjer seg gjeldande både for testserien (éin test per modul) og det individuelle prosjektet som kvar student skal utføre. I motsetnad til opplegget til (Jazayeri, 2015) der ein innleiande meistringsstige med testar deretter kvalifiserte for å delta i eit gruppeprosjekt, tenkjer vi å utføre testserie og prosjekt i parallell, slik at studentane samstundes med at dei jobber med å klare testen for eit visst nivå, også arbeider med å utvikle prosjektet sitt til det nivået (som inneber at prosjektkoden inkluderer føremålstenleg bruk av dei programmeringsmekanismane som inngår i modulen). Ei omtrentleg skisse av innhaldet i kvar modul er gitt i Tabell 1. Bokstavane tilsvarer karakter, dvs. studentane startar på I, og treng å klare nivå I, H, G, F og E for å stå i emnet.

Tabell 1. Skissert modulinndeling av emnet

Nivå	Programmeringskonsept
I	Enkel skjerm-i/o, tal, strengar, variable, tilordning, rekneoperatorar, utføringssekvens
H	Enkle typar, typekonvertering, funksjonsdefinisjonar, lister og arrays (intro), enkel plotting
G	Boolske verdiar, relasjonsoperatorar, if-setningar, f-strengar, indeksring i strengar
F	Lister, tupler, array; løkker (for og while); augmentert tilordning
E	To-dimensjonale data og løkker, meir om funksjonar og skop, enkel fil-i/o (vha. numpy);
D	Skiving; muterande endring, meir om fil-i/o (tekstfiler); meir om strengar, unntak
C	Tilfeldige tal; meir om lister, meir komplekse if-strukturar; mengder, ordbøker
B	Komplekse problem med lister, array, løkker; Binærfiler; Meir om funksjonar
A	Effektivitet, presisjon (unngå avrundingsfeil); Vanskelegare problem over heile pensum

3.2 Undervisningsopplegg

Sidan kvar student kan velje sitt eige tempo gjennom serien av modular (ein ambisiøs student kan til dømes passere første test i veke 34, andre test i veke 35, osv., medan ein mindre ambisiøs student kanskje passerer første test i veke 36, andre i veke 38, ..., vil det ikkje vere slik at alle studentar jobbar med dei same programmeringskonseptane til same tid. Såleis er det ikkje føremålstenleg å ha plenums undervisning av programmeringskonsept frå faglærar, og heller ikkje å ha obligatoriske øvingar som skal leverast til faste tider. Det som derimot vil vere obligatorisk for studentane, er følgjande:

- Deltaking på seminar, der vi vil prøve å setje studentane saman med andre med liknande framdriftsplan.
- Eitt kort møte mellom student og læringsassistent per veke, for å sjekke status og få råd om vidare jobbing. I dette møtet kan studenten sine siste testresultat vere eit nyttig informasjonsgrunnlag, dette vil gjere det lettare for læringsassistenten å sjå kva for konsept studenten eventuelt slit med å forstå, og dermed kunne fokusere på å forklare dette eller vise til relevante læringsressursar, slik at møtet blir målretta.

Noko plenumspresentasjon vil det nok likevel måtte bli frå faglærar, men da fokusert på innleiande informasjon om opplegget i starten av semesteret, pluss innslag av plenumsmotivasjon undervegs både med tanke på programmeringsfaget og det spesielle undervisningsopplegget.

3.3 Vurderingsordning

For å ta ein viss karakter må studenten (a) bestå alle testar til og med denne karakteren i teststigen, og (b) levere eit prosjekt som også har nådd dette nivået. Dvs., for å ta t.d. C, må ein passere testar til og med C-testen, og levere eit prosjekt som held til C. Det er ikkje snitting av test og prosjekt, så t.d. ein student som leverer eit prosjekt til C men berre tar testar til og med E (eller omvendt) får E (altså lågaste av karakterane), ikkje D eller C. Kvar student kan velje kva veke ein vil møte opp for å ta neste test, og likeeins kva veke ein vil vise fram prosjektet sitt for å få det godkjend for eit visst nivå.. Det er 9 nivå / modular – I, H, G, F, E, D, C, B, A. Studenten må ein bestå 5 av desse, altså til og med E, for å ta ståkarakter i emnet, og ytterlegare modular for å ta gradvis betre karakter.

Figur 1 viser korleis meistringsstigen er bygd opp av ein testserie pluss eit prosjekt. Testane vil vere fullt ut autoretta, og ein student må score minst 90% for å stå testen. Autoretting er dels eit pragmatisk val for at det ikkje skal bli for stor arbeidsbyrde på fagstaben med rettarbeid, men også eit pedagogisk verkemiddel. Ved å lage store spørsmålsbankar der spørsmål blir trekte tilfeldig frå gong til gong, kan systemet da leggast opp mest mogeleg transparent ved at studentane kan trene uformelt på dei same testane som dei skal gjere under tilsyn for å få teljande resultat. Autoretting vil da gjere at studentane kan få rask, automatisk tilbakemelding på korleis dei har prestert og kva dei eventuelt misforstår og må lære seg betre – og slike testresultat kan også vere ein basis for samtale med læringsassistent for å få tips om betring.

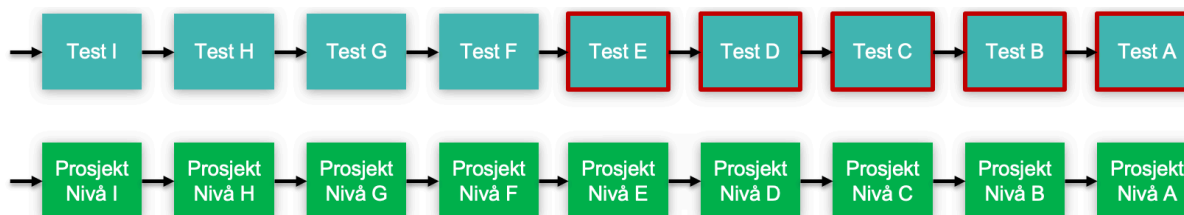


Fig. 1. Meistringsstigen startar på nivå I, og beståtte resultat er frå E til A

Figur 2 viser nokre mogelege studentforløp oppover meistringsstigen. A-45 indikerer ein rask og ambisiøs student som passerer nivå I allereie i veke 44 og tar A i veke 45. B-47 tilsvarande ein student som tar B i veke 47. B-47-UKA viser ein student med same sluttnivå, men med raskare framdrift i starten – for så å gjere lite i vekene 41-44, til dømes på grunn av jobb på studentfestivalen UKA i denne perioden. Og så vidare nedover med lægre ambisjonsnivå og svakare karakter som sluttresultat – men som likevel kan vere eit bra resultat med tanke på realistisk meistring i den aktuelle studenten sin situasjon.

Veke	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
A-45	I	H	G	F	E	D		C		B		A			
B-47		I	H	G		F		E		D		C		B	
B-47-UKA	I	H	G	F		E	D					C		B	
C-47		I		H		G		F		E		D		C	
D-44		I	H	G		F		E			D				
E-40		I	H	G	F		E								
E-46			I		H		G			F			E		
E-48				I		H			G			F			E

Fig. 2. Nokre døme på ulike studentforløp gjennom modulane.

Kvar ein skild test vil ha same innhald for alle studentar – bortsett frå at det vil vere store spørsmålsbankar som det blir trekt tilfeldig frå, slik at ikkje same spørsmål skal gå igjen og igjen ved gjentatte forsøk. Programmeringsprosjektet vil derimot gi mogelegheit for tilpassing til var student sin spesialretning innanfor LUR, og dermed kunne gi ei ekstra kjensle av relevans. Kvar student skal lage eit program som kunne vere brukt i undervisning av vedkomande sine lektorfag, dvs. ein student som går LUR matematikk og fysikk, skal lage eit program som kunne tenkjast brukt i undervisning av eitt eller begge desse faga i vidaregåande skule. Dette vil gi studentane høve til å reflektere over samanhengen mellom programmering og den pedagogiske profesjonen som dei siktar seg inn mot. Samstundes er det verd å merke seg at studentane i første semester på LUR enno ikkje har lært noko teori innan pedagogikk, så ein kan ikkje forvente teoretiske refleksjonar rundt dette – det vil mest måtte vere basert på egne erfaringar som elev.

4 STUDENTINVOLVERING I UTVIKLINGA

Eit 1,5 timars møte med studentar på LUR blei halde 25. oktober 2022. 15 studentar frå ulike årskull av LUR deltok på møtet: 5 frå 2. klasse, éin frå 3. klasse, to frå 4. klasse og sju frå 5. klasse. Studentane representerte ulike fagkombinasjonar. Fleirtalet hadde fysikk og matematikk, men det var òg tre med fagkombinasjonen informatikk og matematikk, to kjemi og matematikk og éin student med biologi og kjemi. Ingen studenter fra matematikk og biologi deltok.

Møtet var interaktivt, og blei innleia med ei digital spørreundersøking med responssystemet Mentimeter. Her tok studentane stilling til ei rekke utsegner om korleis dei sjølve hadde opplevd IT Grunnkurs (men da altså den tradisjonelle varianten med obligatoriske øvingar og avsluttande skriftleg eksamen), med svar på ein Likert-skala. Vidare blei skissa for det nye emnedesignet presentert, i tillegg til at studentane fekk moglegheita til å kome med innspel undervegs. Meir konkret blei studentane presentert for opplegget med testserie og prosjekt i parallell, og korleis dette var tenkt å fungere. Studentane blei delte inn i grupper for å diskutere spørsmål som var laga på førehand, og deretter blei synspunkt som hadde komme fram i gruppene, diskutert vidare i plenum.

Studentane sine tilbakemeldingar kan delast inn i to hovudkategoriar: 1) potensielle utfordringar og 2) generelle meininger. Ei av de potensielle utfordringane studentane tok opp, var at det kan opplevast som stressande og krevjande å måtte ta éin test kvar uke. Dei var også vare for at det kunne vere vanskeleg for førsteårsstudentar å vite kor mykje arbeid som er forventa i starten av semesteret. Det blei også diskutert om prosjektet skulle gjerast i grupper eller individuelt. Somme meinte at det burde vere i grupper fordi studentar, og spesielt lærarstudentar, tidleg i studieløpet burde få trening i samarbeid. Andre meinte at individuelt prosjekt kunne vere betre, både fordi studentar kunne ha ulike førehandskunnskapar og ulike ambisjonsnivå – og dermed ulikt tempo gjennom modulane.

Generelt var studentane positive til dei ideane som blei presentert for det nye emnedesignet, jamvel om dei også uttrykte noko skepsis. Studentane nemnde sjølv at skepsisen kunne vere grunna i at

emnedesignet var såpass nytt og ulikt andre emne dei hadde hatt tidlegare. Dei var svært positive til å få eit eige ITGK-emne berre for LUR-studentane, og at dette emnet ville prøve ut noko nytt for å skape betring.

5 PLAN FOR OPPFØLGING OG FØLGJEFORSKING HAUSTEN 2023

I dette emnet vil det være læringsassistentar som hovudsakeleg deltar inn i tilbakemeldingsprosessen saman med studentane. Det er derfor nødvendig at assistentane blir følgde opp og får innføring og innsikt i relevant teori og rammeverk, og korleis tilbakemeldingsprosessen bør fungere. Læringsassistentar ved NTNU gjennomgår eit lite pedagogisk kurs (LAOS), og den fagspesifikke delen av dette må sannsynlegvis vere noko annleis for studentane som skal medverke på det nye emnet enn det som har vore vanleg for andre læringsassistentar på ITGK.

Av følgjeforskning som skal gjennomførast hausten 2023 er det planar om å bruke både spørjeundersøkingar og fokusgruppeintervju der respondentane er studentane, læringsassistentane og faglærarane. I tillegg vil det også bli gjennomført observasjonar av både fellesøker og rettleiingssamtalar. Formålet vil vere å få fram deltakarane sine opplevingar knytt til den nye praksisen.

6 DISKUSJON OG KONKLUSJON

Emnedesignet for IT1001 IT Grunnkurs kan framstå som svært annleis frå det som elles har vore vanleg ved NTNU – og dermed også som svært innovativt. Som vi har vist i Seksjon 1 og 2, er det likevel ikkje så nytt og innovativt. Bloom og Keller sine idear om meistringslæring er over 50 år gamle, og det fins fleire som har prøvd slike opplegg i innleiande programmeringsemne i nyare tid. Opplegget vi no tenkjer å prøve i IT1001 ved NTNU har mykje sams med fleire av desse andre opplegga som det er referert til. Nokon skilnader er det likevel også. Medan fleire av dei andre opplegga har hatt avsluttande eksamen i tillegg til ein meistringsstige, er vårt opplegg berre fokusert på denne meistringsstigen. Medan dei fleste andre sine meistringsstigar berre består av testar, har vår meistringsstige ein serie testar pluss eit individuelt programmeringsprosjekt som blir utført i parallell med testane. Testar og prosjekt skal sikre to ulike læringsutbytte: testane at studentane har forstått verkemåten til sentrale programmeringskonsept, prosjektet at dei kan skrive samanhengande kode som gjer noko nyttig. Sidan desse studentane går LUR, ei profesjonsutdanning for å bli lektor i realfag i vidaregåande skule, er ei ekstra relevant vinkling på prosjektet at det skal lage kode som kunne vere tenkt brukt i undervisning av realfag – og kvar student skal lage eit prosjekt som passar med sin retning på LUR (dvs., ein student som går LUR matematikk og fysikk må da lage eit program som kunne vere tenkt brukt i undervisning av matematikk og/eller fysikk). Dette prosjektet gjer opplegget noko ulikt det som andre har prøvd. Rett nok hadde også Jayazeri (2015) ein kombinasjon av teststige og prosjekt, men da med teststigen før prosjektet, og som vilkår for å få lov til å ta prosjektet, måtte teststigen vere fullført midtvegs i semesteret (og dei som eventuelt berre fullførte teststigen men ikkje prosjektet, fekk lågaste ståkarakter). Jayazeri sitt opplegg ville altså innebære ein teststige som gjekk opp til karakteren E, og deretter prosjekt for å ta karakterar betre enn E.

Om opplegget i IT1001 blir vellukka, gjenstår å sjå. Mange av dei andre erfaringane som er rapportert med meistringslæring i innleiande programmeringskurs viser at fagstaben har endre opplegget litt frå år til år, basert på at det første året ikkje fungerte heilt tilfredsstillande. Om det skulle bli vellukka, er eit anna interessant spørsmål kor stor ressursbruk emnet krev for å fungere (særleg da timebruk frå faglærarar og assistentar). Om ressursbruken er berekraftig, kan det vere freistande i så fall å prøve å skalere liknande opplegg også til større klassar. LUR er berre 50-100 studentar, medan det er langt fleire på dei andre emnekodane av ITGK: til saman om lag 2500 studentar fordelt på dei tre emnekodane TDT4109, 4110 og 4111. Om det derimot skulle vise seg å krevje mykje meir ressursar, vil det neppe vere aktuelt å prøve å skalere det opp til langt større studenttal.

REFERANSAR

- Alvarez, C., Samary, M. M., & Wise, A. F. (2023). Modularization for mastery learning in CS1: a 4-year action research study. *Journal of Computing in Higher Education*, 1-44.
- Bloom, B. S. (1968). Learning for Mastery. Instruction and Curriculum. Regional Education Laboratory for the Carolinas and Virginia, Topical Papers and Reprints, Number 1. *Evaluation comment*, 1(2), n2.

- Campbell, J., Petersen, A., & Smith, J. (2019). *Self-paced mastery learning cs1*. Paper presented at the Proceedings of the 50th ACM technical symposium on computer science education.
- Dawson, P., Henderson, M., Mahoney, P., Phillips, M., Ryan, T., Boud, D., & Molloy, E. (2019). What makes for effective feedback: Staff and student perspectives. *Assessment & Evaluation in Higher Education*, 44(1), 25-36.
- de Macêdo Morais, L. A., Figueiredo, J. C., & Guerrero, D. D. (2014). *Students satisfaction with mastery learning in an introductory programming course*. Paper presented at the Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE).
- de Pontes, R. G., Guerrero, D. D., & de Figueiredo, J. C. (2019). *Analyzing gamification impact on a mastery learning introductory programming course*. Paper presented at the Proceedings of the 50th ACM technical symposium on computer science education.
- Dweck, C. S., & Master, A. (2012). Self-theories motivate self-regulated learning. In *Motivation and self-regulated learning* (pp. 31-51): Routledge.
- Eyre, H. L. (2007). Keller's Personalized System of Instruction: Was it a Fleeting Fancy or is there a Revival on the Horizon? *The Behavior Analyst Today*, 8(3), 317.
- Garner, J., Denny, P., & Luxton-Reilly, A. (2019). *Mastery learning in computer science education*. Paper presented at the Proceedings of the Twenty-First Australasian Computing Education Conference.
- Hattie, J., & Clarke, S. (2018). *Visible learning: feedback*: Routledge.
- Izu, M., Ng, D., & Weerasinghe, H. (2023). *Mastery Learning and Productive Failure: Examining Constructivist Approaches to teach CS1*.
- Jazayeri, M. (2015). *Combining mastery learning with project-based learning in a first programming course: An experience report*. Paper presented at the 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering.
- Johnson, C., & Molloy, E. (2018). Building evaluative judgement through the process of feedback. In *Developing Evaluative Judgement in Higher Education* (pp. 166-175): Routledge.
- Keller, F. S. (1968). Goodbye, teacher. *Journal of applied behavior analysis*, 1, 79-89.
- Kulik, J. A., Kulik, C.-L. C., & Cohen, P. A. (1979). A meta-analysis of outcome studies of Keller's personalized system of instruction. *American Psychologist*, 34(4), 307.
- LeJeune, N. (2010). Contract grading with mastery learning in CS 1. *Journal of Computing Sciences in Colleges*, 26(2), 149-156.
- McCane, B., Ott, C., Meek, N., & Robins, A. (2017). *Mastery learning in introductory programming*. Paper presented at the Proceedings of the nineteenth australasian computing education conference.
- Molloy, E., Noble, C., & Ajjawi, R. (2019). Attending to emotion in feedback. In *The impact of feedback in higher education: Improving assessment outcomes for learners* (pp. 83-105): Springer.
- Ott, C., McCane, B., & Meek, N. (2021). *Mastery learning in cs1-an invitation to procrastinate?: Reflecting on six years of mastery learning*. Paper presented at the Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1.
- Purao, S., Sein, M., Nilsen, H., & Larsen, E. Å. (2017). Setting the Pace: Experiments With Keller's PSI. *IEEE Transactions on Education*, 60(2), 97-104.
- Sindre, G. (2018). *Introductory IT Courses with Large Classes: Perfect Setting for a PSI Revival?* Paper presented at the Utdanning og didaktikk i IT-fagene, Bodø.
- Stegeman, M. (2019). *A set of exercises and tests for teaching tracing skills using a mastery approach*. Paper presented at the Proceedings of the 19th Koli Calling International Conference on Computing Education Research.
- Treviño, Y. M., & Cavazos, M. R. L. (2018). *Effects of immediate feedback using ICT in a CSI course that implements Mastery Learning*. Paper presented at the 2018 IEEE Frontiers in Education Conference (FIE).
- Winstone, N., & Carless, D. (2019). *Designing effective feedback processes in higher education: A learning-focused approach*: Routledge.