

Meistringslæring: Korleis kan du som faglærer utforme og gjennomføre eit emne med studentstyrt framdrift?

Guttorm Sindre, Gabrielle Hansen, *Excited SFU, IDI, NTNU*

SAMANDRAG: Emnet IT1001 ved NTNU nyttar meistringslæring med ein serie modular, der kvar student kan velje sitt eige tempo og ambisjonsnivå. Emnet har fått positiv respons frå studentane. I kva grad kan eit liknande opplegg med meistringslæring være eigna for andre emne i ulike fagfelt? Utforminga av IT1001 blir brukt som døme, men ulike fag kan ha ulike behov, og det finst mange designalternativ innanfor det som kan kallast meistringslæring. Artikkelen prøver å svare på følgjande spørsmål: (1) Korfor bruke meistringslæring? (2) Kva ulike val har faglærer med tanke på utforming av eit emne med meistringslæring? (3) Kva er ein føremålstenleg prosess for utvikling av eit slikt emne? Som artikkelen viser, finst det mange moglege emnedesign innanfor det breie konseptet meistringslæring, og kvar faglærer må finne eit design som passar for sitt emne. Det er viktig å lage eit opplegg som både er handterleg for studentane og for fagstab og administrasjon med tanke på tid og ressursar, og det er ein god idé å involvere studentar allereie i planlegging av emnet.

1 INNLEIING

Emnet IT1001 er eit innleiande kurs i programmering for studentar på Lektorutdanning i Realfag (LUR) ved NTNU, og har vore gitt to gonger, haust 2023 og 2024. Emnet har ei uvanleg utforming: ingen førelesingar, ingen obligatoriske øvingar veke for veke, ingen avsluttande eksamen. Det einaste som er obligatorisk, er deltaking i seminar der studentane arbeider i grupper – 2x45 minutt per veke, må ha møtt minst 11/14 gonger. Pensum er organisert i ein serie modular, der kvar student kan velje sitt eige tempo, og dermed i stor grad bestemme sin eigen karakter, som er basert på kor mange modular studenten har greidd å ta ved semesterslutt. Studentane si evaluering av emnet har vore positiv. Som Figur 1 viser, var det allereie ved første gjennomføring i 2023 meir enn 50 % (11,4+47,7) som var anten svært nøgd eller nøgd, og i 2024 nesten 90 % (39,5+50,0). Figur 2 viser svar på spørsmål om kor nøgde studentane var med IT1001 samanlikna med andre emne dei tok same semester. I 2023 var det over 60 % som tykte IT1001 var betre, og i 2024 over 70 %.

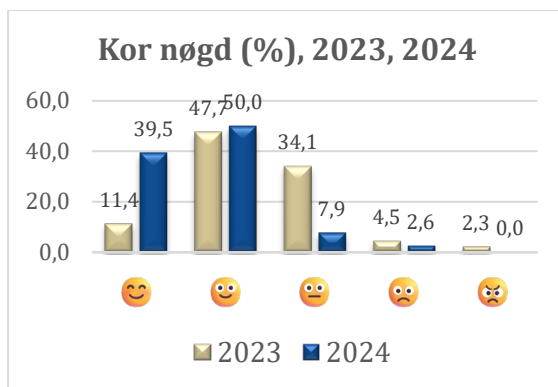


Fig. 1. Kor nøgde studentane var med emnet, '23-24

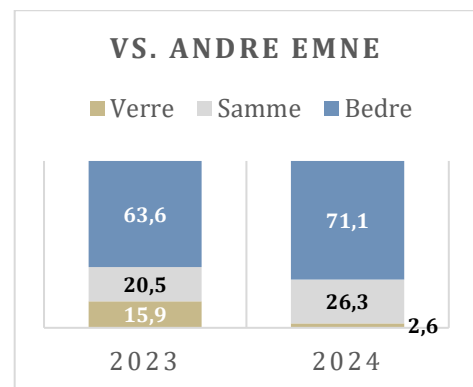


Fig. 2. Nøgde vs. andre emne i same semester

Figur 3 viser karakterfordeling for 2023 og 2024, og dessutan heilt til venstre karakterfordeling for tilsvarande studieprogram i 2022 – men da i emnet TDT4110 som LUR-studentane tok saman med mange andre studentar før dei fekk sitt eige programmeringsemne. Som ein kan sjå, vart karakterane for 2024 vesentleg betre enn i 2023, og dessutan betre enn tilsvarande studentgruppe hadde hatt i 2022, da emnet dei tok var eit vanleg «fast-tempo» emne med førelesingar, obligatoriske øvingar og eksamen på slutten av semesteret. Emnet IT1001 hadde 48 studentar i 2023 og 40 i 2024 – og begge år

var det 0 klager og 0 spørsmål om grunngeving av karakter, trass i eit lågt karaktersnitt spesielt det første av desse åra.

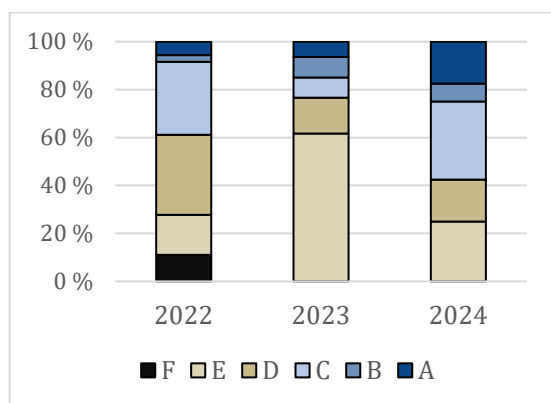


Fig. 3. Karakterar, '22 (eksamen), '23-24 (ML)

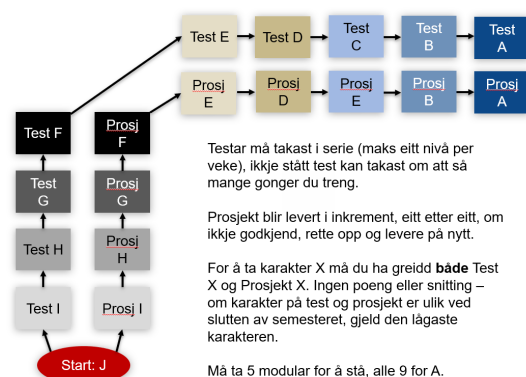


Fig. 4. Meistringsstige med 9 modular, I, ..., A

Figur 4 illustrerer meistringsstigen som utgjør kjernen i emnedesignet. Pensum er delt i 9 modular som har namn i omvend alfabetisk rekkjefølgje I, ..., A. For å ta lågaste ståkarakter E må studenten greie 5 modular I, ..., E, og for å ta A må ein greie alle 9 modulane. I kvar modul dokumenterer studenten meistring av modulen ved å stå ein autoretta test (typisk med stågrense 90 %) og ved å levere ei akseptabel utviding av eit individuelt programmeringsprosjekt. Begge deler måtte gjerast i rekkjefølgje (Test I måtte takast før Test H, osv.; Prosjekt I måtte leverast før Prosjekt H, osv.), men vi kravde ikkje at studentane måtte halde synkront tempo mellom testar og prosjekt (dvs., om ein ville kunne ein godt ta testar I, H, G, ... utan å ha starta på prosjektet, eller levere prosjekt I, H, G, ... utan å ha tatt nokon test) – men *ved slutten av semesteret* måtte ein ha kome fram til same bokstav for test og prosjekt for å ta den karakteren. Kvar student kunne velje sitt eige tempo og ambisjonsnivå innanfor desse rammene. Figur 5 viser klassen sin testprogresjon gjennom semesteret i 2024, nettopp som ein illustrasjon av høgst ulike tempo. På testen i veke 42 var det 8 ulike bokstavar i spel fordi dei kjappaste studentane hadde greidd B-testen veka før og dermed var klare for å prøve seg på A, medan den langsamaste framleis skulle ta H. Når studentane held så ulikt tempo, gir det ikkje mening å ha plenums-førelsing. I dei obligatoriske seminarar jobba studentane i grupper, alt etter kor langt dei hadde kome i meistringsstigen og kva dei helst ville bruke seminartida til – testførebuving eller prosjekt. Figur 6 viser korleis klassen delte seg opp for seminaret i veke 40, 2024.

Det vart gjort nokre endringar i opplegget frå 2023 til 2024, basert på tilbakemeldingar frå studentar og faglærar sine erfaringar – ikkje minst at karaktersnittet i 2023 vart ganske lågt fordi mange studentar valde å stoppe på E. I 2023 sleit mange studentar med å kome i gong med det individuelle programmeringsprosjektet fordi emnet hadde få læringsressursar for prosjektdelen. Det blei derfor laga fleire læringsressursar for dette, pluss nokre kodemalar som studentane kunne nytte for dei første delane av prosjektet. Særleg testane F og E viste seg dessutan å vere i overlag vanskelege i forhold til eit krav om 90% for å stå, og i 2023 blei ein del studentar ståande fast i 3 veker på ein og same test. I 2024 vart det gjort nokre justeringar på dette: (a) det vart lagt inn meir tilbakemelding på somme av testoppgåvene i Inspira, så treningstestar skulle fungere betre som ein læringsressurs i seg sjølv, (b) det blei mogeleg å prøve 2 gonger på same test på den same fredagen, (c) kvar student hadde frå starten av semesteret 3x “80%-sjansar”, altså at somme testar kunne passerast med 80% score heller enn 90. Desse kunne først brukast frå F og oppover (dvs., I, H, G måtte passerast på 90%, og det ville også vere lurt å passere F, E på 90% dersom ein tenkte seg heilt til A). Desse tiltaka gjorde at færre studentar blei ufrivillig ståande fast på same bokstav i mange veker, og gjorde det lettare for studentar å kome over startkneika med å få framdrift i prosjekta sine.

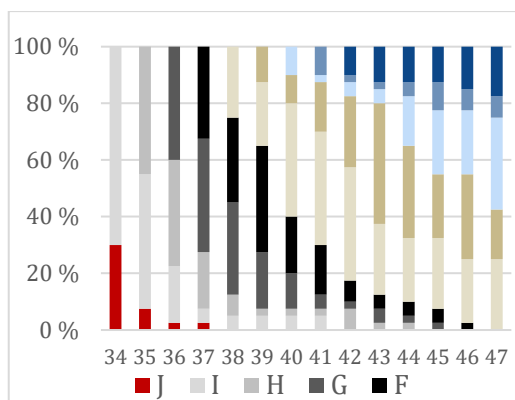


Fig. 5. Testprogresjon gjennom semesteret ('24)

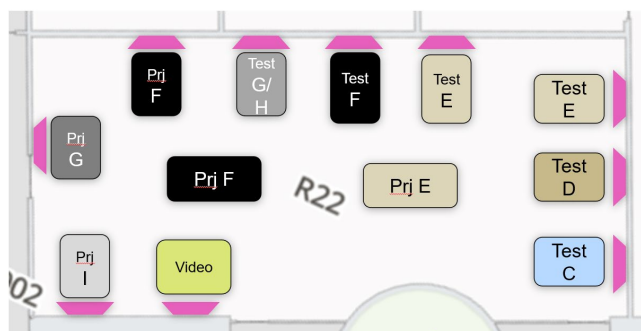


Fig. 6. Seminarrommet, gruppedeling veke 40, 2024

Meir detaljerte skildringar av emnet IT1001 kan finnast i andre kjelder, om tankane bak (Sindre et al., 2023), erfaringar frå første gjennomføring (Hansen et al., 2024; Sindre & Hansen, 2024), og frå andre gjennomføring (Sindre & Hansen, 2025; Sindre et al., 2025). Føremålet med denne artikkelen er i staden å gi inspirasjon til andre faglærarar som kan vere interesserte i å prøve meistringslæring – uansett disiplin, og uansett kor emnet ligg på skalaen praktisk-teoretisk. Spørsmåla vi vil svare på, er desse:

- Korfor bruke meistringslæring?
- Kva ulike val har faglærar med tanke på utforming av eit emne med meistringslæring?
- Kva er ein føremålstenleg prosess for utvikling av eit emne med meistringslæring?

Resten av artikkelen er strukturert som følgjer: Seksjon 2 argumenterer korfor, basert på egne erfaringar pluss litteratur om meistringslæring. Seksjon 3 diskuterer ei rekkje ulike designalternativ som faglærar står overfor, og der ein kan velje å lage eit emne som er utforma om lag på same vis som IT1001, eller kan ende opp med eit ganske annleis emnedesign, alt etter kva som er mest høveleg i den aktuelle konteksten. Seksjon 4 diskuterer prosessen for å utvikle emnet, mellom anna med fokus på studentinvolvering. Seksjon 5 konkluderer artikkelen.

2 KORFOR BRUKE MEISTRINGSLÆRING?

Den sentrale ideen bak meistringslæring som føreslått på 1960-talet (Bloom, 1968; Keller, 1967, 1968) er at studentar lærer i ulikt tempo. Når eit undervisningsopplegg derimot blir køyrt i eitt fast tempo (t.d. gjennom førelesingar eller obligatoriske øvingar veke for veke), vil dette vere for langsamt for somme studentar og for raskt for andre. Dei som fell av lasset, vil ha problem med å hente seg inn att fordi undervisninga går vidare til neste tema før dei har meistra det meir grunnleggjande. Løysinga er derfor å la studentar halde ulikt tempo, det som på engelsk blir kalla *student-paced* eller *self-paced learning* (Tullis & Benjamin, 2011), i motsetnad til opplegg der tempo er styrt av faglærar.

Vårt emne, IT1001, handlar om programmering – men det er ingenting ved meistringslæring som avgrensar bruken til dette spesielle faget. Keller (1967) sitt Personalized System of Instruction (PSI) blei først brukt i psykologi, og ein nyleg studie (Mannion et al., 2023) fann publikasjonar om bruk i undervisning i mange ulike disiplinar, som psykologi, programmering, matematikk, biokjemi, og språkfag. I innleiande programmering på universitetsnivå finst det fleire døme på bruk av meistringslæring der studentane iallfall i nokon grad kan bestemme eige tempo (Garner et al., 2019), likeins i ingeniørfag (Perez & Verdin, 2022). Ein sentral grunn til å bruke meistringslæring er at det har vist god læringseffekt i empiriske studiar (Kulik et al., 1990), og det kan finnast fleire årsaker til dette:

- Hyppig testing, der kvar test har lågare risiko enn éin stor slutteksamen har vist seg å auke motivasjon og læring og redusere angst hos studentar (Bangert-Drowns et al., 1991; Paloposki et al., 2025; Yang et al., 2021). Dette stemmer med meir generelle funn om at kortare tilbakemeldingssyklusar gir betre læring (William, 2018).

- Sjansar til å ta mislukka testar om att, slik at studentar kan betre læringa si og rette opp tidlegare feil, bidrar også til å redusere angst og motivere for vidare læring (Emeka et al., 2023; Feldman, 2020; Fernandez, 2021), og særleg når studenten også kan gjere dette i sitt eige tempo så ein ikkje treng å kome til ein test før ein er klar for det (Bell, 2023).
- Ved å velje sitt eige tempo blir studenten gitt meir kontroll og ansvar for eiga læring, som kan bidra til større grad av sjølvregulering (Bell, 2023; Kulik & Kulik, 1987).
- Det blir tydeleg for studenten korleis hen ligg an fagleg gjennom semesteret, kva som er neste steg, og kva som er kravet for å stå. Dette er noko mange slit med å sjå i «vanlege» undervisningsopplegg, og som da er ei kjelde til frustrasjon og stress. Med eit opplegg som vist i Figur 4, har studenten kontroll til ei kvar tid: «Eg ligg no på F. Det neste eg treng å gjere, er å greie E-testen og levere E-prosjektet» ; «Eg ligg no på D. Dersom eg ønskjer ein betre karakter, må eg greie C-testen.»

Det blir også tydeleg for faglærar korleis både klassen generelt og einskildstudentar ligg an gjennom semesteret. Ved å følgje med i ein oversikt som vist i Figur 5 veke for veke, veit faglærar til ei kvar tid kor mange ein har på ulike nivå. Er det nokon studentar som ikkje har levert noko på prosjektet enno? Er det nokon som har tatt så få testar at det er grunn til uro, eller som gjentatte gonger har feila på same test? Slik innsikt i studentane sin progresjon får ein ikkje i førelesing i eit auditorium, eller ved å sjå at alle har levert obligatoriske øvingar som for somme sin del kan vere kopierte frå medstudentar eller løyste ved hjelp av KI. Sidan testane i meistringslæring derimot skjer under tilsyn, er faren for kok eller KI-bruk liten.

3 ALTERNATIV FOR UTFORMING AV EMNET DITT

Den sentrale ideen i meistringslæring er å dele eit emne opp i modular som skal takast i sekvens, og der studenten skal ha vist meistring av modul 1 før hen går vidare til modul 2, osb. Men det finst mange variasjonar i korleis eit slikt undervisningsopplegg kan utformast. I denne seksjonen vil vi gå gjennom følgjande punkt:

- **Modularisering og sekvensering.** Korleis dele pensum inn i modular, korleis setje modulane i rekkjefølgje?
- **Vurderingsform og karakterar** – formalitetar og praktiske råd.
- **Dokumentering av meistring per modul:** testar, lab, innleveringsarbeid og/eller noko anna?
- **Testar:** Dersom opplegget ditt skal bruke testar, korleis leggje opp desse med tanke på sjansar for øving og gjentak.
- **Prosjekt / semesteroppgåver:** Dersom opplegget ditt skal bruke prosjektarbeid, semesteroppgåver eller andre innleveringsarbeid, korleis kan dette leggjast opp?
- **Læringsressursar:** Kva ressursar må lagast for å gjere det mogeleg for kvar student å halde sitt tempo?
- **Seminar, læringsaktivitetar.** Korleis kan dette leggjast opp når studentar held ulikt tempo gjennom pensum?

Desse spørsmåla blir omhandla i kvar sin delseksjon nedanfor.

3.1 Modularisering, sekvensering

Målet er at kvar modul skal vere liten nok til å ikkje bli overveldande for studenten, og at studenten skal kunne vise meistring av éin modul og deretter gå vidare til den neste. Ideelt sett vil vi da ha mange fingranulære einingar. Samstundes kan det vere arbeidskrevjande med svært mange einingar med ein kontrollert test etter kvar, så praktiske omsyn kan trekkje i retning av færre modular. Tabell 1 nedanfor gir døme på ulike val når det gjeld tal på modular eller læringseiningar. Fred Keller sitt tidlege opplegg i det grunnleggjande psykologi-emnet PY112 som omhandla i den legendariske artikkelen «Goodbye teacher...» står i rad 2.

Tabell 1. Oppdeling i modular eller læringseiningar, ein del døme

Emne	Lærestad	Referanse	Einingar	Kommentar
------	----------	-----------	----------	-----------

Kjemi	Pasadena Comm. College	(Tay, 2024)	47	Del av emne, svært korte testar, berre 2 spm per eining
Psykologi	Arizona State U.	(Keller, 1968)	30	20x lesepensum med test (50%), 10x lab (20%), eksamen (30%)
Matematikk	Brunel U.	(Brinkman et al., 2007)	20	4 modular à 5 einingar
Fysikk	U. Rochester	(Masi et al., 2015)	17	Kvasi-eksperiment, trad. (N=160) vs. meistringslæring (N=151)
IT	UiA	(Purao et al., 2017)	15	Starta med fleire, redusert til 15
IT	UBC, Canada	(Toti et al., 2023)	13	12 test + 1 prosjekt til slutt
Lingvistikk	Georgetown U.	(Fasold, 1974)	13	
IT	George Mason Univ.	(Offutt et al., 2017)	10	Lov å halde fram neste semester
IT	NTNU (vårt emne)	(Hansen et al., 2024)	9	Ingen eksamen, testar + prosjekt, obligatoriske seminar
IT	U. Otago	(Ott et al., 2021)	7	Eksamen (50%) to første år
IT	U. Toronto	(Campbell et al., 2019)	7	

Som Tabell 1 indikerer, er det stor variasjon i talet på einingar / modular, og IT1001 ligg i den lågare enden av dette spekteret med berre 9 modular (eller, om ein tel testar og prosjektinnleveringar separat: 18). Eit større tal kunne ha vore betre med tanke på å unngå at kvar modul skal bli overveldande for studenten, men 9 modular var det vi landa på med tanke på å finne ein balanse mellom kva som var tenleg for å unngå overveldande modular (som kunne ha trekt i retning av fleire), og kva som var realistisk med tanke på tid og ressursar både for studentar og fagstab (som kunne ha trekt i retning av færre modular). Det er også verd å merke seg at eit større eller mindre tal på einingar til dels kan vere eit spørsmål om praktisk organisering snarare enn fundamentale faglege ulikskapar. Kvar modultest i IT1001 består av fleire deloppgåver (typisk 9-10). I teorien kunne vi ha delt opp kvar modul i læringseiningar, t.d. med éi deloppgåve i kvar eining, der studenten måtte vise meistring av oppgåve I-1 før hen kunne gå vidare til I-2, osv., Da ville vi lett ha kome opp i eit tal på einingar tilsvarande det som finst i Kjemikurset til (Tay, 2024). Men dette ville ha blitt krevjande å gjennomføre i praksis. Vi kunne ikkje ha arrangert summative testar under tilsyn for kvar deloppgåve, så det måtte i så fall ha vore berre formativt med automatisk godkjenning online, men sjølv dette ville ha vore tungvint slik LMS og digitalt eksamenssystem ved NTNU fungerer per i dag.

Råd om tal på modular: Det finst ikkje nokon fasit på rett tal modular. Del inn i så mange modular som mogeleg innanfor rammene av logisk samanheng i emnet og passande arbeidsbelastning for studentar og fagstab.

Eit anna spørsmål går på sekvens av modulane, der Figur 7 illustrerer tre ulike paradigme. Somme emne er bygde opp slik at det neste som kjem, er gradvis meir avansert og alltid byggjer på det som er presentert tidlegare. Da er gjerne rekkjefølgja gitt av seg sjølv: tema 2 byggjer på tema 1 og danner grunnlag for tema 3, rekkjefølgja må derfor vere 1, 2, 3. Vi kallar dette vertikal sekvens, som vist med grøn farge til venstre i Figur 7: første modul startar med det enklaste, og det blir deretter gradvis meir avansert. I andre tilfelle kan det vere at det ikkje finst noka slik opplagt rekkjefølgje i eit pensum fordi det dekkjer ei rekkje ulike tema. Dette er kalla horisontal sekvens og er vist midt i Fig.7 med blå farge – her kunne kanskje 3, 2, 1 ha vore ei like naturleg rekkjefølgje som 1, 2, 3 – vi må berre gjere eit val. Dei fleste emne er ein mellomting: Dei dekkjer fleire ulike tema, men samstundes er det visse deler som byggjer på andre delar, med ein del avanserte element som ein først bør gi seg i kast med når ein har skjøna det grunnleggjande. Ein mogeleg sekvens for dette er skissert til høgre i Figur 7, der ein først dekkjer emnet sine ulike tema på eit grunnleggjande nivå, deretter går gjennom tilsvarande tema på meir avansert plan. Eksakt utforming kan variere – eit breitt emne der studentane ikkje skal opp på særleg avansert nivå, kan t.d. ha 12 modular i form av 6 horisontale tema og berre 2 vertikale nivå, medan eit smalare emne der studentane skal ha meir avansert kunnskap, kan ha 12 modular arrangert i 3 tema og 4 vertikale nivå. Dessutan treng ikkje heile matrisa vere full – t.d. er kanskje det aller mest avanserte nivået berre med i pensum for eitt av tre tema, medan dei to andre kjem i eit seinare emne.

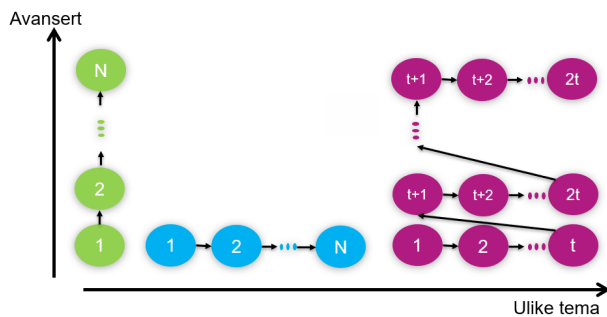


Fig. 7. Ulike paradigmer for sekvensering: vertikal (venstre), horisontal (midt), blanda (høgre)

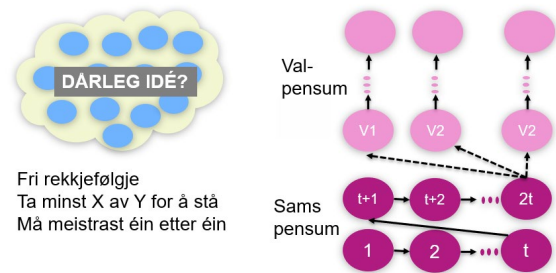


Fig. 8. Alternativ til rein sekvens? Fri rekkjefølgje (venstre), sams pensum + valpensum (høgre)

Særleg dersom ein har eit emne som passar best til den blå varianten midt i Fig. 7 (horisontal sekvens), kan ein spørje om éin fast sekvens er naudsynt? Vil det ikkje gi enda meir ansvar (og sjølvregulering) til studenten om modulane kan takast i akkurat den rekkjefølgja studenten sjølv ønskjer, som indikert i Figur 8, venstre del? Dette meiner vi likevel vil vere ein dårleg idé av minst tre grunnar: (i) utforming av læringsressursar og modultestar, (ii) fagleg-sosialt fellesskap i klassen, og (iii) arbeidsmengd for fagstaben. Med tanke på punkt (i), ta eit tenkt døme, vi har eit emne der studentane skal lære om planetane i solsystemet. Reint overordna er mange rekkjefølgjer mogeleg, frå inst til ytst (Merkur, ...), frå nærmast oss sjølve og vekk (Jorda, ...), frå størst til minst (Jupiter, ...), men når vi først har valt ei rekkjefølgje, vil vi typisk ønskje å definere nokre omgrep relatert til planetar i tidlege modular (t.d. kva er ein planet, kva er ein måne, eksentrisitet, omløpstid, ...) for så å kunne bruke desse omgrepa i læringsressursane for seinare modular. Om studentane skulle kunne velje rekkjefølgje på modular heilt fritt, må alle modulane innehalde like mykje forklaring til desse omgrepa. Studenten får da heller inga kjensle av progresjon frå den eine modulen til den neste, rett nok blir stadig nye planetar omhandla, men alle blir presenterte på nybyrjarnivå. (ii) Med meistringslæring med sjølvvald tempo blir det etter kvart eit skilje mellom raskare og langsamare studentar i klassen – men sjølv dei langsamaste har likevel nokon å arbeide saman med så lenge det finst andre som er like langsame, fordi alle skal starte på same modul. (t.d. i det tenkte astronomikurset, om alle skal byrje med Merkur og det fem veker inn i semesteret er tre studentar som enno ikkje har kome i gong, så er dette likevel 3 som alle treng å lære om Merkur, medan valfri rekkjefølgje lettare kunne føre til at nokon vel å starte på ein modul som ingen andre held på med. (iii) Etter kvart må fagstaben tilpasse seg at studentar held på med mykje ulikt, til dømes ved at ein må tilby ulike testar den same veka fordi somme studentar har kome langt, andre kort, men med sams rekkjefølgje er det iallfall slik at faglærar i starten kan konsentrere seg om undervisning og godkjenning av den første modulen, så det blir lettare å få etablert klassen og opplegget i starten av semesteret.

Råd om modulsekvens: Organiser modulane i ei gitt rekkjefølgje, jamvel i situasjonar der denne rekkjefølgja ikkje er strengt naudsynt.

Det kan likevel tenkjast situasjonar der ikkje alt treng å vere i ein rein sekvens. Som døme, lat oss tenkje oss eit emne i forskingsmetodar for ei brei gruppe av studentar, der det er ønskeleg at alle studentar skal få ei generell innføring i eit breitt spekter av forskingsmetodar, men dessutan også at kvar student skal kunne få meir avansert innsikt i den eller dei forskingsmetodane som vil vere mest aktuelle for vedkomande sitt masterprosjekt. Ein kunne da tenkje seg eit opplegg som vist til høgre i Figur 8, med eitt sams pensum med modular som dekkjer ulike metodar (mørk raud) og etter dette modular som går meir i detalj på spesifikke metodar, der studentar kan velje eitt eller fleire slike tilvalsspor alt etter personleg interesse. Vi har ingen erfaringar med eit slikt emnedesign, men det er mogeleg å tenkje seg dette, særleg om det er ein ganske stor klasse der det likevel blir mange nok på kvar spesialisering mot slutten av semesteret.

3.2 Vurderingsform og karakterar

I formelt byråkratispråk fell vurderingsforma i IT1001 i sekkeposten «Mappevurdering», sjølv om det ikkje er det som ein pedagogisk ville kalle mappevurdering (der studenten kan bestemme kva som skal vere med i mappa), men snarare det vi vil kalle ein meistringsstige. Fire designval som du kan ta her:

1. Kva karaktersystem skal brukast (A-F eller Bestått / Ikkje bestått)?
2. Kor mykje tel meistringsstigen av totalkarakteren?
3. Korleis blir prestasjon på meistringsdelen vurdert: talet på modular tatt, eller oppsamla poeng frå modular?
4. Kva grad av fridom har studenten til å velje sitt eige tempo?

Karaktersystem: IT1001 nyttar A-F, og dei aller fleste andre opplegg slik som kjeldene i Tabell 1 nyttar også graderte karaktersystem. Det ville ha vore enkelt å gjere om IT1001 til Bestått / Ikkje bestått, måtte i så fall velje kor mange modular måtte takast for Bestått (T.d. 5 som for E? 6 som for D?). Det vi ser for oss ville vere ei utfordring, vil vere å motivere studentar for å ta modular ut over det som er kravd for Bestått.

Råd om karaktersystem: Så lenge dei andre emna som studentane tar parallelt i det same semesteret har bokstavkarakter, bør også ditt emne ha bokstavkarakter, sidan Bestått / Ikkje bestått lett vil gi redusert innsats i emnet.

Del av totalkarakteren: I IT1001 tel meistringsstigen 100/100, det finst ingen andre vurderingsdelar som bidrar til karakteren. Dette er ikkje det mest vanlege for emne med meistringslæring rundt om kring i verda. Dei fleste av døma i Tabell 1 inkluderer ein slutteksamen. Til dømes hadde det banebrytande opplegget til Keller (1968) 50 % vekt på modultestar, 20 % på lab-øvingar og 30 % på avsluttande eksamen. Ott et al. (2021) hadde avsluttande eksamen dei to første åra (50 %) dei prøvde ut eit opplegg med meistringslæring, men deretter gjekk bort frå dette og baserte karakter på meistringstestar (60 %) og to praktiske semesterprøver til fastsette tidspunkt (40 %).

Råd om eksamen: Unngå slutteksamen om du kan! Særleg dersom meistring av modulane blir dokumentert ved testar under tilsyn, blir eksamen smør på flesk og vil medføre mykje ekstra arbeidsinnsats både for faglærar og studentar. Ein eksamen som berre tel 20-30 %, er mykje herk for liten karaktereffekt, og dersom han tel meir, vil opplegget gå glipp av den mogelege vinsten ved redusert eksamensstress for studentane. Dessutan kan ein eksamen på fast dato i slutten av semesteret vere ei kilde til prokrastinering (Sindre & Hansen, 2025). Dette fordi eksamen heilt til slutt hindrar studentar i å gjere seg ferdig med emnet tidlegare, og dersom dei uansett må ha heile pensum friskt i minnet ved semesterslutt, kan det kjennest like lurt å ha tatt mesteparten av modulane ganske nært opp til eksamen. Ein mogeleg motivasjon for å ha eksamen kunne vere dersom dokumentasjon av meistring per modul *ikkje* er testar under tilsyn, men derimot prosjektarbeid, semesteroppgåve eller anna som skjer utanfor tilsyn, og der ein kan vere usikker på om arbeidet er gjort av studenten sjølv eller med hjelp frå venner eller KI. Etter vår meining er likevel ei betre løysing på ei slik uro å putte inn testar under tilsyn som del av meistringsstigen, heller enn å putte inn ein avsluttande eksamen.

Råd om andre delar av mappa: Berre éin del, meistringsstige 100/100, er eit reint og enkelt design – vurder dette om du kan! Om ein skal ha inn andre komponentar som må takast til faste tider heller enn å vere opp til kvar student, bør det vere gode faglege eller praktiske årsaker til dette – sidan aktivitetar som berre er mogeleg å få gjort på eitt fast tidspunkt, bryt med idéen om at studentar skal kunne velje eige tempo – særleg dersom aktiviteten krev at studenten har oppnådd eit visst kunnskapsnivå som dei langsamaste kanskje enno ikkje har. Dessutan vil tidfesta aktivitetar som tel på karakteren, lett gi administrative komplikasjonar med å handtere eventuelt sjukefråvær hos studentar. Ein god fagleg grunn til tidfesta aktivitetar kan likevel vere dersom emnet inneheld lab, feltarbeid, praksis eller liknande som av omsyn til HMS, tilgang til avansert utstyr, fagstab eller eksterne aktørar ikkje kan tilbydast kvar veke men må skje til visse tidspunkt i semesteret. I så fall kan det vere fornuftig å ta dette ut av meistringsstigen og ha det som ein eigen del som tel X %, der storleiken på X vil vere avhengig av kor mykje av emnet sitt læringsutbyte som er knytt til den tidfesta aktiviteten.

Vurdering av prestasjon på meistringsdelen. IT1001 reknar kvar modul berre for stått / ikkje stått, typisk med ei stågrense på 90 % på test, 100 % på prosjektrubrikk, og karakter blir bestemt av talet på modular som studenten har stått (5 modular for E, ..., 9 modular for A). Eit liknande design er også brukt av (Toti et al., 2023), der karakter i all hovudsak er bestemt av kor mange modular studenten har tatt – vel å merke med sjanse til å score nokre få ekstrapoeng på aktiv deltaking i klassen, som er meir vanleg å bruke i Amerika enn i Noreg. Mange andre emnedesign innan meistringslæring bruker i staden oppsamla poeng frå modulane til å bestemme karakter, og slik at alle studentane skal ta alle modulane, heller enn at ein berre treng å ta alle for å nå toppkarakteren. Eitt døme er opplegget til (Offutt et al., 2017). Der måtte alle studentane ta alle 10 modulane for å få ståkarakter, men stågrensa

per modul var lågare (70 %) enn det som er vanleg for meistringslæring (gjerne 90 %). Ein student som berre så vidt sto kvar modul på 70 %, ville typisk da hamne på lågaste ståkarakter i emnet, medan ein student som scora monaleg over stågrensa ville kunne samle opp poeng til betre karakter.

Personleg er vi svært nøgde med opplegget for IT1001. Når kvar modul er berre stått/ikkje stått, utan noka oppsamling av poeng, slepp studentar å dra med seg dårlege prestasjonar frå tidlege testar vidare utover i semesteret. Ein føresetnad for at dette skal fungere, er nok at emnet er delt opp i mange nok modular, og særleg at vegen opp til lågaste ståkarakter er delt opp i mange modular (for IT1001: I, H, G, F, E). Om eit emne hadde hatt berre 5 modular, som med eit opplegg som gav karakter per modul da måtte ha vore for E, D, C, B, A, ville innhaldet i den første modulen ha blitt altfor omfattande og overveldande for ein del studentar, og ein ville ikkje ha oppnådd dei fordelane som meistringslæring har i form av oppdeling av stoffet.

*Råd om koplinga modular – karakterar: Dersom du har kapasitet til å ha mange nok modular (minst 8, gjerne fleire) i emnet ditt, er teljing av kor mange modular studenten har stått, eit reine vurderringsdesign enn å samle poeng for modulane. Særleg i det norske universitetssystemet er det viktig å ha eit vurderringsdesign som **i minst mogeleg grad inviterer til klager**.* Det vil potensielt vere temmeleg kostbart med to klagesensorar som skal vurdere test svar eller innleveringsarbeid for N modular. Modular som berre er stått / ikkje stått, utan noka poengstraff for veik prestasjon eller repeterte forsøk, er meir tenleg for å unngå klager enn modular der studenten skal samle opp poeng som etter kvart skal bli til ein karakter. Over dei to åra IT1001 har køyrt, har det vore 0 klager og 0 spørsmål om grunngjeving av karakter. Rett nok berre med 88 studentar til saman, men gitt at dei fleste emne har meir enn 10% klager og meir enn 20% grunngjevingar, er 0 likevel påfallande.

Studenten sin fridom til å velje tempo. IT1001 har få avgrensingar på studenten si framdrift, men heilt fritt er det likevel ikkje. Av omsyn til kapasitet har kvar student berre sjanse til å stå éin teljande test kvar veke. Ein svært rask student, som til dømes kan mesteparten av programmeringspensum frå før, må dermed likevel bruke minst 9 veker på å kome seg til A, og ein student som har låge ambisjonar og helst vil vente så lenge som mogeleg, må likevel ta I-testen seinast i veke 10 (og deretter greie H, G, F, E i vekene 11-14) for å rekke å stå i emnet.

Det finst nokre få døme på emne som er mindre restriktive enn dette. Den originale idéen til Keller (1968) var at studentar som trong meir tid, kunne halde fram inn i neste semester. Eit liknande opplegg blei brukt i programmeringskurset til Offutt et al. (2017), der studentar som ikkje rakk å ta alle modulane innan semesteret var over, kunne halde fram inntil 10 veker inn i neste semester. Dette er heilt i tråd med ideala bak meistringslæring: Alle studentar kan oppnå meistring berre dei får lang nok tid på seg. I eit nokså firkanta norsk universitetssystem er dette urealistisk å få til, så for IT1001 er emnet slutt når semesteret er slutt, og studentar kan ikkje gå på overtid inn i neste semester. Sidan studentane da skal i gong med nye emne, ville ein kanskje ha gjort dei ei bjørneteneste om ein tillét å bruke meir tid. Dessutan skal også faglærarar i gong med undervisning av nye emne.

Dei fleste andre emne med meistringslæring er snarare *meir restriktive* når det gjeld studenten sin fridom til å velje framdrift. Det finst to vanlege motivasjonar for dette: (i) praktiske omsyn til fagstaben sin kapasitet, og (ii) motverke studentar sin tendens til å prokrastinere. Hos Toti et al., (2023) er motivasjonen primært kapasitet. Med ein ganske stor klasse (ca. 300 studentar) var det avgrensa kor mange ulike testar dei kunne arrangere samstundes, og kor mange alternative oppgåver fagstaben hadde tid til å lage til kvar modultest (ved feila test, kunne studentane ta testen om att neste veke – men da ville det bli for lett om dei fekk heilt lik oppgåve). Dei landa på ei løysing med 3 variantar av kvar test, og der kvart testnivå dermed berre var tilgjengeleg i eit tre vekers tidsvindaug – altså kunne dei langsamaste studentane maksimalt liggje tre modular bak dei raskaste (medan IT1001 hadde ei spreiding på 6 modular i 2023 og 8 modular i 2024). Sjølv om denne avgjerda primært var av omsyn til kapasitet, kan det også motverke prokrastinering ved at studentar får ein siste frist for å ta innleiande modular allereie tidleg i semesteret. I mange emnedesign finst det også spesifikke tidsfristar eller tidfesta aktivitetar med tanke på å redusere prokrastinering. Ott et al. (2021) fortel om utviklinga av eit emneopplegg ved Univ. Otago, New Zealand over ein femårsperiode, der fleire av tiltaka som blei gjort etter kvart, var motivert ut frå reduksjon av prokrastinering. Likeeins fortel (Purao et al., 2017) om utviklinga av eit opplegg ved UiA over fleire år, der endringar år for år dreidde seg mykje

om å motverke prokrastinering. Tiltak mot prokrastinering kan typisk vere siste fristar for å ta tidlege modular eller vurderingsaktivitetar som berre er tilgjengeleg på fast tidspunkt i semesteret.

Om vi skulle hatt meir tiltak mot prokrastinering i IT1001, kunne dette typisk ha vore å setje ein siste frist for å ta tidlege modular – slik som at I-modulen må vere stått seinast veke 5, H seinast veke 7, G seinast veke 9 (i staden for 10, 11, 12 som er de-facto fristar no, om ein skal rekkje E innan semesterslutt). Dersom vi hadde opplevd store problem med prokrastinering, ville vi nok ha gjort noko slikt, men det har ikkje vore så ille. Som vi kan sjå av Figur 5, som viser testprogresjon for 2024-klassen, så er den gråsvarte «halen» av studentar som framleis heldt på med testar lågare enn E seint i semesteret, ganske tynn. Den studenten som venta aller lengst, var i ein situasjon der hen måtte stå testane G, F, E i veke 12, 13, 14 for å stå – men greidde også dette. Det var fleire studentar som viste ei åtferd motsett av prokrastinering, nemleg å prøve å gjere seg ferdig tidleg – slik som 4 studentar som tok 9 testar på rad dei 9 første vekene. Dei fleste emne med meistringslæring som har hatt problem med prokrastinering, har også hatt avsluttande eksamen som ein del av opplegget. Ein spekulasjon kan derfor vere at det ikkje er fridom til å velje tempo i seg sjølv som skaper prokrastinering hos mange studentar, men slik fridom i kombinasjon med avsluttande eksamen – som gjer at det uansett er umogeleg å gjere seg ferdig med emnet tidleg, og for mange da kan opplevast som opportunt å leggje mest innsats seint i semesteret for å ha kunnskapen frå modulane friskt i minne til eksamen.

Råd om studentar sin fridom til å velje tempo: Dersom omsyn til kapasitet tilseier det, eller den aktuelle klassen har låg motivasjon og store utfordringar med prokrastinering, kan det vere greitt å setje grenser for studentane sitt val av tempo – men ikkje overdriv dette! Å la studentar velje sitt eige tempo er ein sentral idé i meistringslæring og ein nøkkel til å promotere betre sjølvregulering hos studentane, som i nokon grad vil gå tapt med mykje fristar og avgrensingar. Tidsfristar kan også føre til administrativt meirarbeid for faglærarar, til dømes dersom sjukefråvære eller andre legitime årsaker til at studentar kan vere forseinka fører til at ein må gi dispensasjon og lage ekstraordinære vurderingsaktivitetar.

3.3 Dokumentere meistring per modul

Som vist i Figur 4 har IT1001 testar og prosjekttinnlevering i parallell for kvar modul. Dette er eit meir komplekst emnedesign enn det som er vanleg – meir normalt ville t.d. vere å ha ein meistringsstige med berre testar. Det er likevel ein grunn til at vi har valt å gjere det slik:

- Testane er 100 % autoretta med oppgåvesjangrar tilgjengeleg i Inspira, som er det systemet som blir brukt for digital eksamen ved NTNU. Dette blir dermed fleirval, innfyllingsoppgåver, dra-og-slepp-oppgåver der kodeliner skal puslast saman i rett rekkjefølgje, etc. – men ikkje oppgåver som testar skriving av kode. Om dette hadde vore oppgåver med kodeskriving i slike kjappe testar, ville det likevel også berre ha blitt korte kodesnuttar med svært spesifikke krav, ikkje meir kreativ programmering.
- Det individuelle programmeringsprosjektet gir derimot studentane mogelegheit til å skrive eit større program relatert til deira eigen fagdisiplin. Studentane på LUR går ulike spesialiseringar allereie frå første semester, t.d. matematikk + fysikk eller kjemi + biologi. Prosjektet er ope, kvar student kan lage kva hen vil – bortsett frå at det må vere eit program som er tenkt brukt i samband med undervisning av studenten sitt disiplinfag (t.d. naturfag, kjemi) i ungdomsskule eller vidaregåande. Prosjektet tar dermed emnet inn i ein kontekst av studenten sin hovuddisiplin, og dessutan også med ein relasjon til studenten sitt framtidige yrke som lektor, ved at prosjektet skal resultere i eit program som kunne vere tenkt brukt til undervisning i eit skulefag.

Dei fleste emne med meistringslæring har enklare meistringsstigar der det berre er testar, t.d. (Campbell et al., 2019; Offutt et al., 2017; Ott et al., 2021). Somme har ein serie testar og så eit prosjekt heilt på slutten (Jazayeri, 2015; Toti et al., 2023), men da blir prosjektet berre tilgjengeleg for dei studentane som kjem lengst i emnet, ikkje for heile klassen. Eit anna alternativ er å ikkje ha testar, slik at kvar modul i staden blir vurdert ved innleveringsarbeid. Dette blir gjort i eit innleiande programmeringsemne ved UiA (Purao et al., 2017), der studentane må forklare koden sin munnleg til

assistentar ved kvar innlevering for å redusere risiko for at nokon leverer arbeid dei ikkje har laga sjølve. Dessutan har emnet ved UiA ein munnleg eksamen i slutten av semesteret.

Alt i alt kan ein sjå for seg følgjande alternativ: berre testar (som kan vere skriftlege, munnlege, eller praktiske, alt etter kva det aktuelle emnet har behov for), berre innleveringsarbeid (semesteroppgåver / øvingar / lab-rapportar / prosjekt / ...), testar og innleveringar i parallell (som IT1001 har), testar og innleveringar i alternerande sekvens, eller testar og deretter innleveringar mot slutten – til dømes berre for studentar som siktar høgare enn E. Her gjeld det å finne ein balanse mellom kva som er mest i tråd med emnet sine læringsutbytte, og kva ein har kapasitet til å gjennomføre – pluss som nemnt tidlegare: unngå alternativ som inviterer til mykje klager på karakter.

I dei to neste seksjonane vil vi skrive meir om testar og innleveringar, sidan dette er dei to elementa som vi sjølve har erfaring med i IT1001.

3.4 Testar

Testane i IT1001 har berre autoretta oppgåver. Dette har fleire fordelar. Studenten kan sjå sekundet etter at svar er levert i Inspira om det vart nok poeng til å stå eller ikkje, og treng dermed ikkje vente på noko rettarbeid frå fagstaben for å finne ut dette. Hen kan også kjapt finne ut kva deloppgåver som hadde feil svar. Autoretting har også fordel med å ikkje invitere til klager, i motsetnad til testar der fagstaben må bruke skjønn. Unnataket ville vere høve der det var feil i fasiten til ein test, og ein student dermed var blitt snytt for akkurat det eine poenget vedkomande mangla for å stå testen. I IT1001 blei slike situasjonar (ein handfull i 2023, berre to i 2024) korrigert av faglærar innan to timar etter testen, heller enn at det skulle trenge å føre til klager i ettertid.

Råd om testar: Bruk autoretta testar, ikkje manuelt retta testar der det trengst skjønn i vurderinga – med mindre du har læringsutbytte som gjer slike skjønnsbaserte testar heilt naudsynte.

Testane i IT1001 blei brukte både formativt og summativt – slik at studentane kunne trene på dei same testane som dei ville få under tilsyn som teljande test. Dette er uvanleg – til dømes hadde nemnte opplegget til (Toti et al., 2023), som elles likna noko på IT1001, berre tre variantar av kvar testoppgåve, og desse var da forståeleg nok ikkje tilgjengelege for trening på førehand. For å kunne tilby formativ trening på same oppgåvebank utan å ende opp med enkel pugg av svar, måtte faglærar i IT1001 lage ganske mange variantar av kvar oppgåve. Typisk ville kvar modultest innehalde om lag 10 deloppgåver, kvar med 20 ulike variantar som Inspira trekte tilfeldig frå for kvar student ved kvar gjennomføring av ein formativ treningstest eller summativ test under tilsyn. Denne doble bruken av testane (formativ + summativ) var godt likt av studentane fordi det gav ei kjensle av kontroll og gradvis meistring. Studentar kunne gjennomføre ein treningstest og sjå på scoren sin – t.d. 60 % - medan den summative testen ville trenge 90 % for å stå. Ved å undersøke kva deloppgåver som førte til poengtap kunne studenten få klare indikasjonar på kva som måtte lærast betre, og prøve å forstå dei konseptane som førebels var misforstått anten ved å sjekke løysinga / forklaringa i treningstesten direkte, eller ved å konsultere digitale læringsressursar (notatbøker, videoar) for dei aktuelle deloppgåvene. Ved deretter å gjere nye forsøk på treningstestar, kunne studenten sjå gradvis framgang – kanskje 75 %, 85 %, 92 % – og etter kvart konkludere med at hen var klar for ein teljande test. Forsking indikerer også at hyppig bruk av formative vurderingar gjer det lettare for studentar å oppleve meistring (Hansen & Ringdal, 2018). Øving på treningstestar vart ein sentral aktivitet i seminara i IT1001, der studentar på same nivå sat i grupper med testen oppe på ein veggskjerm og diskuterte seg gjennom løysing og fasitsjekk.

Råd om formativ-summativ bruk av testar: Dersom det finst kapasitet til å lage mange nok variantar av oppgåver, bruk testar som ein formativ læringsressurs i tillegg til summativt. Dersom ein berre har kapasitet til å lage forholdsvis få oppgåver, må ein halde dei variantane som skal brukast i summative testar separat frå det som blir gjort tilgjengeleg for trening, slik at studentane ikkje blir eksponerte for dei summative oppgåvene på førehand.

Kva som er god utforming av testoppgåver, vil variere avhengig av kva type emne det er snakk om (disiplin, nivå, type av læringsutbytte), og kva oppgåvetypar som er støtta i det systemet som ein vel å bruke for testane. Testane i IT1001 var i Inspira, og vi brukte ei rekkje ulike oppgåvetypar, som fleirval, paring, nedtrekk, plasser-i-tekst, fyll-inn-tekst, dra-og-slepp. Med den kombinerte formativ-summative bruken av testoppgåver var det viktig å ha tilstrekkeleg variasjon mellom ulike variantar av

same oppgåve. Figur 9 og 10 nedanfor viser éin av dei 21 variantane av oppgåve G-9. Teksten øvst, rutenettet til høgre og kodelinene som skulle plasserast, var dei same kvar gong, men orda Søtt, Fett, OK og Rart var plasserte på ulike måtar i 2x2-tabellen nedst i Fig.9, som medførte at kodelinene måtte plasserast i ulike rekkjefølgjer og strukturar i rutenettet til høgre for å få rett svar.

G-9 Skript med valg

Vi skal fullføre eit Python-skript der dei to første kodelinene allereie er gitt:

```
a = input("Tegn 1: ")
b = input("Tegn 2: ")
```

Etter dette skal dei neste kodelinene ta nokre avgjerder om printing av tekst basert på om `a == "A"` eller om `b == "B"` eller ikkje. Tabellen under viser dei fire moglege køyringane, basert på alle kombinasjonar teikna som kom inn, var A og B eller ikkje.

	<code>a == 'A'</code>	<code>a != 'A'</code>
<code>b == 'B'</code>	Tegn 1: A Tegn 2: B Søtt Fett	Tegn 1: V Tegn 2: B Søtt Fett
<code>b != 'B'</code>	Tegn 1: A Tegn 2: y OK Rart Fett	Tegn 1: V Tegn 2: y OK Fett

Trekk kodelinene nedanfor til rett plass så koden vil verke som gitt i alle dei fire tilfella over, når

Fig. 9. Ein variant av testoppgåve G-9

```
if b != "B":
    if a == "A":
        else:
            print("Søtt")
            print("OK")
            print("Rart")
            print("Fett")
```

Fig. 10. Kodeliner og innfyllingsfelt for G-9

Tilbakemelding

FASIT:

```
if b != "B":
    print("OK")
    if a == "A":
        print("Rart")
    else:
        print("Søtt")
    print("Fett")
```

TIPS: Her er det viktig å vurdere nøye logisk operator i puslebiten (er det == lik, eller != ulik) og sekvens av ord i utskrift i tabellen. Er if-setn. avhengige (andre if nosta inni første) eller uavheng (begge if på venstre marg)? Skal else med første eller andre if?

FORKLARING HER: Ordet Rart er kun i ruta nederst venstre og er dermed avhengig begge if- dvs. her må vi ha nøsting av if inni if. Søtt står kun i øverste rad av tabellen (dvs. n "B"), OK står derimot kun i begge ruter når b != "B", så Søtt vs. OK har if-else-relasjon. Den b-te puslebiten vi har her, er b != "B", vi må da starte med denne for å printe OK, så neste inn a-teste printe Rart, deretter ta else for å printe Søtt. Viktig at else skal med b-testen, må altså ut på ytter marg igjen. Ordet Fett står sist i alle fire ruter og er dermed uavhengig av begge if-setn., skal de bli printa nederst på ytterste marg.

Fig. 11. Tilbakemelding delvis korrekt svar, G-9

Fig. 12. Forklaring etter fasit, G-9

3.5 Innleveringar (prosjekt / semesteroppgåver / lab-rapportar / ...)

Med innleveringar meiner vi her arbeid som ikkje er gjort under tilsyn, men som skal vere med på å dokumentere at studentar har meistra ein modul. I IT1001 hadde vi individuelt prosjektarbeid i tillegg til testane. Vi valde individuelt prosjekt heller enn gruppeprosjekt fordi gruppearbeid kunne bli problematisk når studentar held svært ulikt tempo gjennom emnet – til dømes at studentar med låge ambisjonar lett blir gratisspassasjerar om dei kjem i gruppe saman med nokon med høge ambisjonar. Eit anna element var at kvar student skulle kunne velje sitt eige tema for prosjektet ut frå fagdisiplin og interesse, så éin student kunne ønskje å lage eit quiz-program for Biologi 1 vgs., medan ein annan kunne ønskje å lage ein simulator knytt til eit fysikk-eksperiment med ballkast og luftmotstand. Med gruppearbeid ville det lettare bli at somme ville ende opp med eit prosjekttema som det primært var einkvan annan som var interessert i. Med individuelt prosjekt kan alle ta det temaet dei aller helst vil ha, dermed større eigarskap. Sidan IT1001 er eit nybyrjarkurs i programmering, verkar det dessutan fornuftig å la kvar student finne si eiga indre røyst i skrivning av programkode før ein gir seg i kast med den auka kompleksiteten i eit gruppearbeid. Dette treng likevel ikkje tyde at gruppeprosjekt er ein dårleg idé i eit emne med meistringslæring, men ein må i så fall tenkje ut eit opplegg der ein anten koplar studentar med same tempo og ambisjonsnivå, eller har eit rammeverk for å mogeleggjere at ulike personar kan bidra inn i eit prosjekt med ulike tempo og ambisjonsnivå.

Det finst døme på opplegg som berre har innleveringar per modul, heller enn testar – slik som programmeringskurset ved UiA (Purao et al., 2017). Her er det kodeskrivingsoppgåver som studentane skal utføre og levere per modul, der dei må forklare koden sin for assistentar i samband med leveringa. Avhengig av fagdisiplin og læringsutbyte kan innleveringar vere nesten kva som helst, som t.d. tekstlege dokument (semesteroppgåver, refleksjonsnotat), podcasts, videopresentasjonar, kunstverk, modellar. Nett som for testar, tenkjer vi at det er ein fordel om vurderinga av slike innleveringar i

minst mogleg grad inviterer til klager, og at det såleis er tenleg at det finst klare rubrikkar for å vurdere det som er levert, og at det helst er berre stått / ikkje stått – der studenten må levere på nytt med manglar retta opp dersom det blir «ikkje stått» – heller enn at ein samlar opp poeng som skal bli til ein karakter.

I IT1001 heldt studenten på med det same prosjektet heile semesteret igjennom, men med gradvis utviding av programkoden. For kvar innlevering var det ei sjekkliste med krav som studenten måtte fylle inn som sjølvvurdering, og som faglærar deretter sjekka om stemde. I utgangspunktet skulle alle punkta ha svar Ja, og dersom dette var rett, ville innleveringa bli godkjent. Om ikkje godkjent (sjeldan) ville faglærar gi tilbakemelding om korfor, og studenten kunne fikse dette og levere på nytt. Krava til prosjektleveringane i IT1001 kan verke litt opp-ned, ved at det primært er krav til kva programmeringsmekanismar som må vere brukte i koden, ikkje krav til kva programmet skal gjere. Dette heng saman med at prosjektet var tematisk ope der kvar student kunne bestemme sjølv kva programmet skulle gjere, så det var umogeleg å gi faste krav til programmet si åtferd. Idéen var dessutan at prosjektleveringa skulle dokumentere at studenten meistra praktisk bruk av dei same programmeringsmekanismarane som den tilsvarande testen hadde handla om, dermed vart det tenleg å rette vurderingsrubrikken mot korrekt bruk av visse programmeringsmekanismar per modul.

Tabell 2. Sjekkliste for G-prosjektet IT1001

#	Spørsmål	Ja	Nei
1	Koden køyrer feilfritt		
2	Programmet gir forståeleg informasjon ut på skjermen		
3	Koden er delt opp i funksjonar		
4	Koden inneheld minst éi if-else-setning		
5	Koden inneheld minst éi if-elif-else-setning		
6	Koden har minst eitt logisk uttrykk med relasjonsoperatorar (>, <, >=, <=, ==, !=)		
7	Koden har minst eitt uttrykk med logiske operatorar (and, or, not)		
8	Koden har minst éin variabel av type bool		
9	Koden har minst éin bruk av indeksering av streng		
10	Koden utfører minst éi tilfeldig trekking med <code>random.randint()</code> eller <code>random.choice()</code>		

I ditt emne kan det vere annleis med tanke på akkurat kva rubrikken bør fokusere på, avhengig av type emne og studentane sitt nivå – men uansett er det ein fordel å ha ein tydeleg rubrikk der det er lett å forklare korfor noko ikkje er blitt godkjent, og der fagstaben klarer å gi tilbakemelding rimeleg kjapt, slik at studentar har sjansen til å rette opp, levere på nytt og kome seg vidare til neste modul.

Råd om innleveringar: Dersom emnet bruker innleveringar for å dokumentere meistring av modular, legg vekt på tydelege krav som mogeleggjjer både sjølvvurdering frå studentane si side og rask respons frå fagstaben. Om ein bruker berre innleveringar, ikkje testar under tilsyn, er det også viktig å utforme innleveringane eller vurderinga av dei på ein slik måte at ein reduserer risikoen for at studentar får godkjent arbeid som dei ikkje har utført sjølve (men t.d. fått / kjøpt av venner eller oppnådd ved illegitim bruk av KI).

3.6 Læringsressursar

Når studentar held svært ulikt tempo gjennom emnet, er det lite tenleg å presentere fagstoff i plenum i frontale førelesingar – da vil somme studentar ha kome for kort og ikkje ha føresetnad for å skjønne det som blir presentert, og andre har allereie kome lenger og vil oppleve det som keisam repetisjon. I IT1001 var alle læringsressursar digitale. Kvar modul hadde ein serie med Jupyter Notebooks og

videoar som forklare programmeringskonsept og gav tips til korleis dei kunne brukast i prosjekt. Dei formative treningstestane som nemnt tidlegare var også ein viktig læringsressurs.

Råd om læringsressursar: Sats på digitale læringsressursar strukturert per modul.

3.7 Seminar, læringsaktivitetar, læringsareal

Relatert til det ovannemnde punktet kan ein tenkje at det blir vanskeleg å ha nokon fornuftig seminaraktivitet i eit emne der studentane held mange ulike tempo. Samstundes er det viktig for studentar å treffast, ikkje berre i fritida men også i fagleg-sosiale samanhengar. Utan nokon synkron aktivitet der studentar arbeider saman, kan eit emne med meistringslæring i individuelt tempo lett bli eit svært einsamt opplegg der kvar student sit aleine med sine digitale ressursar og ikkje anar kor langt resten av klassen har kome. IT1001 er det einaste emnet som heile klassen av LUR-studentar har saman i første semester, så det har også ein viktig sosial funksjon i form av å skape klassemiljø og kjensle av å høyre til. Emnet inkluderer dessutan læringsutbyte knytt til samarbeid: «*Kan demonstrere og forklare eigen kode for andre og gi konstruktive tilbakemeldingar på andre sin kode*»¹ Vi har derfor obligatorisk oppmøte i seminar (minst 11/14 for å få karakter), men om ein skal ha obligatorisk oppmøte, er det også viktig at seminaret har eit innhald som studentane finn verdfullt. Som vist i Figur 6, brukte vi eit interaktivt læringsareal (R22), og studentar blei grupperte saman med dei som var på same nivå, avhengig av om dei ville jobbe med prosjekt eller testførebuing i den aktuelle økta. Dei som jobba med testførebuing, ville typisk ta opp ein treningstest på ein veggskjerm og så diskutere seg gjennom denne saman, sjekke score etterpå og prøve å forstå eventuelle feil, deretter kanskje prøve éin gong til dersom det var tid igjen. Dei som jobba med prosjekt, kunne diskutere sams problemstillingar, som til dømes «Har du nokon idé til kva du skal bruke if-elif-else-setning til i G-prosjektet?» eller «Eg får ikkje plottet mitt fram på skjermen, er det nokon annan som har hatt liknande problem?» Sidan dei sat saman med andre på same nivå, var det meir sannsynleg at dei hadde liknande faglege utfordringar i den aktuelle timen. Faglærer og læringsassistentar var også til stades i seminar, slik at studentar også kunne få hjelp med problem som ingen medstudentar hadde svar på. Seminara fungerte såleis som eit læringsfellesskap der studentaktivitet og dialog sto i sentrum, og der studentar kunne lære både frå kvarandre og frå fagstaben. Det gav også godt høve for faglærer å bli kjent med studentane og sjå kva dei deler av stoffet dei hadde utfordringar med.

I teorien kunne det ha vore mogeleg å ha seminar utan at faglærer var til stades, til dømes berre med læringsassistentar for å svare på spørsmål frå studentane, men sidan det ikkje var nokon førelesingar, var det naturleg for faglærer å vere til stades på seminar, som gav faglærer betre innsikt i studentane sitt faglege nivå og stemninga i klassen. Tidlegare forskning indikerer også at det er viktig for studentane si oppleving og motivasjon for å delta i formative vurderingsaktivitetar (som studentane si gjennomføring av treningstestar i seminar er eit døme på) at faglærer viser genuin interesse for å medverke i desse aktivitetane (Hansen, 2020). I IT1001 var det rett nok obligatorisk oppmøte på seminar, men det kan likevel vere stor skilnad på om studentar kjem berre for å få registrert oppmøte eller om dei kjem for å vere aktive, og det sistnemnde var for det meste tilfelle i IT1001.

Det er viktig å ha eit læringsareal som er tenleg for den aktiviteten som skal utførast. Dersom målet er at studentar skal kunne samarbeide og diskutere, bør arealet innby til dette (Andersen & Korpås, 2018). For vår del var det dessutan viktig at mange av gruppeborda hadde skjermar på veggane, som gjorde det lettare for mange studentar saman å jobbe seg gjennom dei formative treningstestane, og slik kombinasjon av studentaktive læringsareal og digitale hjelpemiddel kan vere nyttig i mange samanhengar (Andersen & Korpås, 2022). Det finst mange ulike faktorar å tenkje på når det gjeld undervisning i denne typen areal (Korpås et al., 2023), og dersom ein ikkje har erfaring med slik undervisning og slike areal tidlegare, kan det vere lurt å konsultere personar med ekspertise på dette for å diskutere seg fram til kva areal som vil passe til dei aktivitetane som det er ønskjeleg at studentane skal utføre.

Råd om seminar: Eit emne med meistringslæring bør ha seminar eller andre typar læringsaktivitetar der studentane får sjansen til å samarbeide og diskutere, og vurder gjerne å ha obligatorisk oppmøte

¹ <https://www.ntnu.no/studier/emner/IT1001#tab=omEmnet>

på dette, særleg for førsteårsstudentar som treng å bli kjende med andre i klassen. Om faglærar har kapasitet til det, er det ein klar fordel om faglærar sjølv er til stades i seminara, heller enn at det berre er assistentar. Det er også viktig å ha eit eigna læringsareal for aktivitetane, og dersom ein har lite erfaring med undervisning i slike areal tidlegare, kan det vere lurt å konsultere ekspertise for å kome fram til kva areal som vil vere mest eigna, pluss å ha noko ekstra støtte særleg i dei første øktene.

3.8 Bruk av læringsassistentar

I Keller (Keller, 1967, 1968) sitt PSI-opplegg frå slutten av 60-talet spelte assistentar («tutors») ein sentral rolle. Det var assistentar som køyrde testar for studentane og hadde diskusjonar med studenten i etterkant, og andre assistentar som var involverte i lab-øvingane, medan faglærar si oppgåve . Stort behov for assistentar har da også vore ein kritikk mot PSI, med tanke på kostnader og ressursbruk. Som Eyre (2007) påpeiker, kan moderne IT-støtta læring avhjelpe noko av dette. I IT1001, med autoretta testar, var det ikkje behov for at assistentar skulle involvere seg i køyring og retting av testar. Hovudoppgåva til assistentane blei å vere til stades i seminara, som nemnt i seksjon 3.7 – for å hjelpe studentar anten dei jobba med prosjekt eller testførebuing. I 2023 hadde assistentane dessutan i oppgåve å sjå gjennom prosjektleveringane til studentane, med 7 studentar per assistent. Dette var mogeleg i 2023 fordi emnet som ein førstegongs pilot hadde langt fleire assistentar (7 x 100 t) enn det som ville vere norm for eit emne med om lag 50 studentar. I 2024, da trongare økonomi ved instituttet gjorde det naudsynt med store kutt i assistentar til alle emne, fekk IT1001 berre 2 assistentar à 50 timar, altså berre ein sjuandedel av kva vi hadde hatt året før. Da vart assistentane si oppgåve redusert til berre å hjelpe til i seminara, medan faglærar sjølv såg gjennom studentane sine prosjektleveringar. Det er verd å merke seg at seminara likevel fungerte heilt fint, jamvel om det berre var tre personar for å hjelpe til der (to assistentar pluss faglærar) medan det året før hadde vore åtte (sju assistentar pluss faglærar). Sidan seminaret var organisert i grupper av studentar på same nivå, kunne dei også hjelpe kvarandre.

Råd om assistentar: Det er mogeleg å drive meistringslæring med sjølvbestemt tempo med same tal på assistentar som meir tradisjonelle øvingsbaserte emne typisk har. Ved å ha mest mogeleg autoretting av testar eller andre leveringar kan assistentar heller brukast i seminar eller liknande aktivitetar med studentaktiv læring.

4 PROSESS FOR UTVIKLING AV EMNET DITT

Utviklinga av IT1001 var eitt av delprosjekta som FUL (Forvaltningsorganet for Lektorutdanning) ved NTNU sette i gong for å betre kvaliteten på lektorutdanningane sine i tråd med råd som kom fram i NOKUT si evaluering av lektorutdanningane i Noreg^{2,3}. Det var derfor noko økonomisk støtte frå FUL til å førebu emnet, og dette blei brukt til å involvere studentar – meir spesifikt masterstudentar ved LUR, som dermed var godt kjende med studieprogrammet og korleis LUR-studentane opplevde førstesemesteret og den dåverande versjonen av IT Grunnkurs. Det er verd å merkje seg at det ikkje var noka vesentleg misnøye med IT Grunnkurs slik dei hadde det før – dei fleste syntest det var eit greitt emne – men det var rom for betring både i dette emnet i seg sjølv og ikkje minst i korleis det kunne spele ein meir positiv sosial rolle i første semester av LUR-studiet dersom dei fekk sitt eige emne – sidan klassen på det tidspunktet ikkje hadde noko emne felles i første semester.

Studentane var leigd inn i roller som assistentar i førebuing av emnet og bidrog på ulike måtar: (i) organisere møter med LUR-studentar i lågare årskurs for å lufte idear som faglærar hadde om emnedesignet og få innspel frå studentane om ønskje for emnet, (ii) gi tilbakemelding på testoppgåver og læringsressursar etter kvart som faglærar kom i gong med å lage dette. Dette studentbidraget var til stor hjelp i utviklinga av emnet, og studentane er medforfattarar på artikkelen (Sindre et al., 2023).

²https://www.nokut.no/globalassets/nokut/rapporter/ua/2022/sluttrapport-fra-nokuts-evaluering-av-lektorutdanning-for-trinn-813_15-2022.pdf

³https://www.nokut.no/globalassets/nokut/rapporter/ua/2025/oppfolging-av-nokuts-evaluering-av-lektorutdanning-for-trinn-813_3-2025.pdf

Våre råd for utvikling av eit emne med meistringslæring er som følgjer:

1. Involver studentar tidleg, slik at ein får viktige innspel før avgjerder blir tatt om kva som skal stå i studiehandboka (t.d., læringsutbyte, vurderingsform). Dersom det er eit emne i høgare årskurs, kan det vere mogeleg å involvere dei studentane som faktisk skal ta emnet. For eit emne i første årskurs, er dette ikkje mogeleg, men eit godt alternativ er å involvere eldre studentar i same studieprogram, som kjenner studiet og vil ha gode innsikter i kva som kan vere tenlege løysingar.
2. List opp alt det du ønskjer at studentane skal lære i emnet (på ein langt meir detaljert måte enn typiske skildringar av læringsutbyte i studiehandboka) og prøv å fordele det i følgjande kategoriar:
 - a. **MÅ:** heilt naudsynt kompetanse som studenten skal ha etter emnet, dvs. ein student som ikkje kan alt det som er lista i denne kategorien, kan ikkje få ståkarakter og studiepoeng i emnet.
 - b. **BØR:** kompetanse som studentar også normalt skal ha frå emnet, men som dei kan sleppast gjennom utan å ha. Dvs., ein student kan få ståkarakter ved å tilfredsstille MÅ men ikkje BØR.
 - c. **KAN:** kompetanse som ein ville forvente at dei beste studentane har skaffa seg frå emnet, men kanskje ikkje fleirtalet.
3. Del opp MÅ-kompetansen i mindre modular. Dersom du tenkjer å bruke bokstavkarakterar i emnet ditt, vil summen av MÅ-kompetansen utgjere det som trengst for å ta karakteren E. Emnet IT1001 har delt dette inn i 5 modular (I, H, G, F, E) – du kan gjerne ha fleire eller færre alt etter kva som passar ditt emne sine behov, men eit generelt råd er at det bør ikkje vere for få. Å ha heile E-pensumet berre som éin modul, vil gjere denne modulen stor og uhandgripeleg for studentane. Ein fordel med å ha mange modular på vegen mot E er at den første modulen kan gjerast ganske enkel slik at studentar flest får ein positiv start på emnet. Dvs., for kvar modul frå startmodulen til og med det som trengst for E, spesifiser i detalj kva denne modulen skal innehalde, slik at desse modulane til saman dekkjer MÅ-kompetansen.
4. Spesifiser korleis studentane kan demonstrere kompetansen som ligg i kvar modul innanfor MÅ-delen. Er til dømes alt saman testar? Og/eller innleveringar? Lab? Målet er å velje vurderingsformer som passar til læringsutbyta, men samstundes kome fram til eit design som er enkelt å forstå for studentane, greitt å gjennomføre for fagstaben, og som ikkje inviterer til klager.
5. Del opp BØR-kompetansen og KAN-kompetansen i modular tilsvarande. BØR-kompetansen vil hamne i modular som trengst for å ta D og C, KAN-kompetansen i modular for B og A. IT1001 har her berre éin modul per bokstav, men det er sjølvsagt mogeleg også å ha fleire dersom du finn dette tenleg. I førebuingane til IT1001 lurte vi ei stund på om vi skulle ha to modular per karakter over E, altså I, H, G, F, E, E+, D, D+, C, C+, B, B+, A – som ville ha gitt 13 modular. Vi slo frå oss dette både av omsyn til fagstaben (redd at det ville bli for mykje arbeid) og studentane (ein student som var ute etter A, ville da berre ha éi veke slakke i eit semester på 14 veker – og det skjer jo lett at ein kan vere sjuk eller på annan måte forhindre nokon gonger i løpet av eit semester). Ein annan refleksjon er at eit slikt opplegg lett ville føre til at fleire studentar ville velje å stoppe på E, sidan det deretter ville vere to nye testar (og for IT1001 også to nye prosjekttinnleveringar) for å kome opp til neste karakter. Uansett er det vegen fram til E som det er viktigast å dele opp i fleire etappar.
6. Lag læringsressursar og vurderingar for modulane, frå botnen og oppover. Her er det lurt å ha studentassistentar som kan prøve ut ressursane og vurderingsformene og gi tilbakemeldingar.
7. I utforming av emneskildringa er det også lurt å vere i kontakt med administrasjon / eksamenskontor for å vere sikker på at det ikkje er nokon juridiske eller administrative skjær i sjøen med tanke på korleis du tenkjer å designe emnet.

Det ovanståande er sjølvsagt berre ei rådgevande skisse – det finst nok mange andre måtar å gå fram på, og mange ting som inngår i utvikling av eit emne men som ikkje er nemnt over fordi det er innforstått. I tillegg til råda over, kan det sjølvsagt også vere lurt å gjere eit litteratursøk: Finst det nokon ved andre universitet som allereie har brukt liknande emnedesign innan din fagdisiplin? Korleis delte dei pensumet sitt inn i modular? Kva vurderingar brukte dei for modulane. Kva erfaringar har dei gjort seg – til dømes endringar frå første til seinare gjennomføringar av emnet fordi første gong ikkje vart heilt vellukka?

5 KONKLUSJON

Med utgangspunkt i erfaringane våre frå utvikling og gjennomføring av emnet IT1001 Informasjonsteknologi grunnkurs for førsteårsstudentar ved Lektorutdanning i realfag (LUR) ved NTNU har vi i denne artikkelen prøvd å svare på tre spørsmål: (1) Korfor bruke meistringslæring? (2) Kva ulike val har faglærar med tanke på utforming av eit emne med meistringslæring? (3) Kva er ein føremålstenleg prosess for utvikling av eit slikt emne?

Når det gjeld korfor, finst det fleire motivasjonar: Aktivisere studentane, gi dei større ansvar og kontroll over eiga læring og redusere stress i eksamensperioden ved at iallfall eitt av emna på det tidspunktet allereie har resultert i ein karakter. Det gir også meir informasjon til faglærar undervegs i semesteret, ved at ein kan sjå klassen sin progresjon veke for veke og eventuelt bli merksam på studentar som slit eller treng ekstra oppfølging – heller enn at ein under eksamensretting ved juletider blir negativt overraska over kor lite somme studentar kan.

Når det gjeld design av slike emne, finst det som vist i artikkelen ein god del variasjon i designmåtar. IT1001 har valt eit design med testar og innleveringar i parallell per modul, og der talet på modular studenten har tatt, direkte avgjer karakteren. Men det finst mange døme på emne som har gjort annleis – til dømes berre testar, berre innleveringar, eller tillegg av lab-aktivitetar, og der det eine kjem etter det andre heller enn parallelt. Det finst også døme på opplegg der studentane samlar poeng som til slutt blir karakter heller enn at kvar modul berre er stått/ikkje stått, og ein god del døme på emne som gir noko mindre fridom til studenten i å velje eige tempo enn det som er tilfelle for IT1001.

Når det gjeld det tredje spørsmålet om utviklingsprosess for emnet, er det mindre å finne i litteraturen sidan dei fleste publikasjonar fokuserer på forskingsmetode og resultat meir enn korleis sjølve emnedesignet gradvis vart til. Vi har prøvd å gi nokre råd om prosess i seksjon 4, men her må du nok i stor grad finne din eigen veg. Uansett er det lurt å vere ei gruppe faglærarar og studentar som samarbeider om utvikling av emnet, gjerne med innspel frå administrativt tilsette og pedagogisk støttepersonell. Vi kan gjerne også vere med som diskusjonspartnarar dersom nokon har lyst til å prøve meistringslæring med eit opplegg som liknar på det som IT1001 har.

REFERANSAR

- Andersen, T. H., & Korpås, G. S. (2018). Studentaktiv læring i interaktive læringsarealer gjennom lærersamarbeid. *Læring om læring*(1). <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/2594393>
- Andersen, T. H., & Korpås, G. S. (2022). A qualitative study on how to scaffold for collaborative learning in an innovative learning area, a student perspective. *Uniped*, 45(2), 142-152. <https://doi.org/10.18261/uniped.45.2.6>
- Bangert-Drowns, R. L., Kulik, J. A., & Kulik, C.-L. C. (1991). Effects of frequent classroom testing. *The Journal of Educational Research*, 85(2), 89-99. <https://www.jstor.org/stable/27540459>
- Bell, E. C. (2023). Self-pacing and multiple-attempt assessment to address student anxiety in pharmacy calculations. *Discover Education*, 2(1), 9. <https://link.springer.com/content/pdf/10.1007/s44217-023-00032-3.pdf>
- 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. <https://eric.ed.gov/?id=eD053419>
- Brinkman, W.-P., Rae, A., & Dwivedi, Y. K. (2007). Web-based implementation of the Personalised System of Instruction: A case study of teaching mathematics in an online learning environment. *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT)*, 2(1), 39-69. <https://www.igi-global.com/article/international-journal-web-based-learning/2978>
- Campbell, J., Petersen, A., & Smith, J. (2019). *Self-paced mastery learning CSI* Proceedings of the 50th ACM Technical Symposium on Computer Science Education, <https://doi.org/10.1145/3287324.3287481>
- Emeka, C. A., Zilles, C., West, M., Herman, G. L., & Bretl, T. (2023). *Second-Chance Testing as a Means of Reducing Students' Test Anxiety and Improving Outcomes* 2023 ASEE Annual Conference & Exposition, <https://peer.asee.org/44207>
- 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. <https://psycnet.apa.org/fulltext/2008-05986-007.html>

- Fasold, R. W. (1974). Teaching linguistics by a personalized system of instruction. In F. P. Dineen (Ed.), *Linguistics: Teaching and Interdisciplinary Relations* (pp. 43-57). Georgetown University Press.
<https://repository.digital.georgetown.edu/handle/10822/555467#page=55>
- Feldman, J. (2020). Taking the stress out of grading. *Educational Leadership*, 78(1), 14-20.
<https://eric.ed.gov/?id=EJ1267593>
- Fernandez, O. E. (2021). Second chance grading: An equitable, meaningful, and easy-to-implement grading system that synergizes the research on testing for learning, mastery grading, and growth mindsets. *Primus*, 31(8), 855-868. <https://www.tandfonline.com/doi/full/10.1080/10511970.2020.1772915>
- Garner, J., Denny, P., & Luxton-Reilly, A. (2019). *Mastery learning in computer science education* Proceedings of the Twenty-First Australasian Computing Education Conference,
<https://doi.org/10.1145/3286960.3286965>
- Hansen, G. (2020). Formative assessment as a collaborative act. Teachers' intention and students' experience: Two sides of the same coin, or? *Studies in Educational Evaluation*, 66, 100904.
<https://doi.org/https://doi.org/10.1016/j.stueduc.2020.100904>
- Hansen, G., & Ringdal, R. (2018). Formative assessment as a future step in maintaining the mastery-approach and performance-avoidance goal stability. *Studies in Educational Evaluation*, 56, 59-70.
<https://doi.org/https://doi.org/10.1016/j.stueduc.2017.11.005>
- Hansen, G., Sindre, G., Korpås, G. S., & Vassbakk, J. P. (2024). La kvar student velje sitt eige tempo: Meistringslæring i intro programmering. *Læring om læring*, 11(1), 1-13.
<https://www.ntnu.no/ojs/index.php/lo/article/view/5876>
- Jazayeri, M. (2015). *Combining mastery learning with project-based learning in a first programming course: An experience report* 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering,
<https://ieeexplore.ieee.org/document/7202980>
- Keller, F. S. (1967). Engineering personalized instruction in the classroom. *Revista Interamericana de Psicología/Interamerican Journal of Psychology*, 1(3).
<https://journal.sipsych.org/index.php/IJP/article/view/445>
- Keller, F. S. (1968). Good-bye, teacher.... *Journal of applied behavior analysis*, 1(1), 79-89.
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1310979/>
- Korpås, G. S., Røren, K. A. F., & Haugdal, F. (2023). *Evalueringsrapport A2 Øya helsehus-læringsareal for gruppebasert undervisning*.
- Kulik, C.-L. C., & Kulik, J. A. (1987). Mastery testing and student learning: A meta-analysis. *Journal of Educational Technology Systems*, 15(3), 325-345. <https://doi.org/10.2190/FG7X-7Q9V-JX8M-RDJP>
- Kulik, C.-L. C., Kulik, J. A., & Bangert-Drowns, R. L. (1990). Effectiveness of mastery learning programs: A meta-analysis. *Review of educational research*, 60(2), 265-299.
<https://journals.sagepub.com/doi/abs/10.3102/00346543060002265>
- Mannion, A., Coyne, R., Ferrari, C., Neşeli, M., McGee, C., Mollaoglu, S., & Leader, G. (2023). Personalized System of Instruction in Higher Education: A Systematic Review. *Journal of Behavioral Education*, 1-41. <https://link.springer.com/article/10.1007/s10864-023-09530-8>
- Masi, B., Watson, D. M., Bodek, A., Khaitan, D. A., & Garcell, E. (2015). *Comparison of mastery learning and traditional lecture-exam models in a large enrollment physics course* 2015 ASEE Annual Conference & Exposition, <https://peer.asee.org/comparison-of-mastery-learning-and-traditional-lecture-exam-models-in-a-large-enrollment-physics-course.pdf>
- Offutt, J., Ammann, P., Dobolyi, K., Kauffmann, C., Lester, J., Praphamontipong, U., Rangwala, H., Setia, S., Wang, P., & White, L. (2017). *A novel self-paced model for teaching programming* Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale, <https://sparc.cs.gmu.edu/pubs/SPARC-LatS.pdf>
- Ott, C., McCane, B., & Meek, N. (2021). *Mastery learning in cs1-an invitation to procrastinate?: Reflecting on six years of mastery learning* Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1, <https://doi.org/10.1145/3430665.3456321>
- Paloposki, T., Virtanen, V., & Clavert, M. (2025). From a final exam to continuous assessment on a large Bachelor level engineering course. *European Journal of Engineering Education*, 50(1), 164-177.
<https://doi.org/10.1080/03043797.2024.2334728>
- Perez, C., & Verdin, D. (2022). *Mastery learning in undergraduate engineering courses: a systematic review* 2022 ASEE Annual Conference & Exposition, <https://peer.asee.org/40522>
- 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. <https://ieeexplore.ieee.org/document/7553543>
- Sindre, G., & Hansen, G. (2024). *Mastery Learning in Introductory Programming: Running a Project Alongside a Test Ladder* 20th International CDIO Conference, Tunis, Tunisia. https://ntnuopen.ntnu.no/ntnu-xmloi/bitstream/handle/11250/3168026/CDIO_2024_Sindre_Hansen.pdf?sequence=1
- Sindre, G., & Hansen, G. (2025, June 27-July 2, 2025). *A High-Transparency Approach to Mastery Learning in CS1: Mitigating Procrastination Despite Liberal Self-Pacing* In Proceedings of the 30th ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2025), Nijmegen, The Netherlands. <https://doi.org/10.1145/3724363.3729023>

- Sindre, G., Hansen, G., Korpås, G. S., Kirknes, A., Skøien, J. X., & Magnussen, J. L. (2023). Meistringslæring i intro programmering: plan for eit nytt emne. *Læring om læring*, 10(1).
<https://www.ntnu.no/ojs/index.php/lo/article/view/5087>
- Sindre, G., Hansen, G., Korpås, G. S., & Vassbakk, J. P. (2025, 5-6 mai). *Meistringslæring i MNT-fag: Kompetanse på ulike plan* MNT-konferansen 2025, Trondheim.
https://www.ntnu.no/documents/1358741858/1369576868/112_final.pdf/eb30cc77-e18a-e7a7-f55b-fe73a6639955?t=1745414682641
- Tay, G. C. (2024). Personalized System of Instruction for the Foundational Knowledge of Organic Chemistry. *Journal of Chemical Education*, 101(6), 2381-2388. <https://doi.org/10.1021/acs.jchemed.3c01089>
- Toti, G., Chen, G., & Gonzalez, S. (2023). *Teaching CS1 with a Mastery Learning Framework: Impact on Students' Learning and Engagement* Proceedings of the 2023 Conference on Innovation and Technology in Computer Science Education V. 1, <https://doi.org/10.1145/3587102.3588844>
- Tullis, J. G., & Benjamin, A. S. (2011). On the effectiveness of self-paced learning. *Journal of memory and language*, 64(2), 109-118. <https://doi.org/10.1016/j.jml.2010.11.002>
- Wiliam, D. (2018). Feedback: At the heart of—but definitely not all of—formative assessment. In A. A. Lipnevich & J. K. Smith (Eds.), *The Cambridge handbook of instructional feedback* (pp. 3–28). Cambridge University Press. <https://psycnet.apa.org/record/2018-28959-001>
- Yang, C., Luo, L., Vadillo, M. A., Yu, R., & Shanks, D. R. (2021). Testing (quizzing) boosts classroom learning: A systematic and meta-analytic review. *Psychological bulletin*, 147(4), 399.
<https://psycnet.apa.org/record/2021-24563-001>