

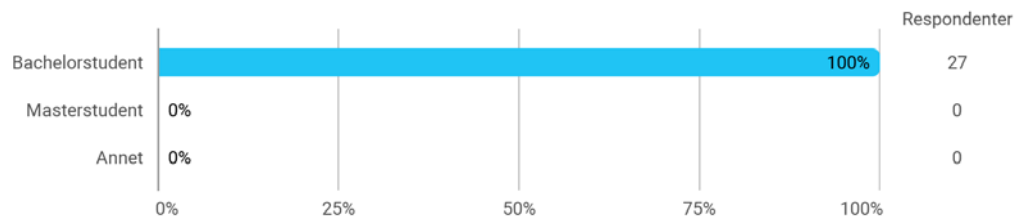
Tilbakemelding

inf-122, h-2018.

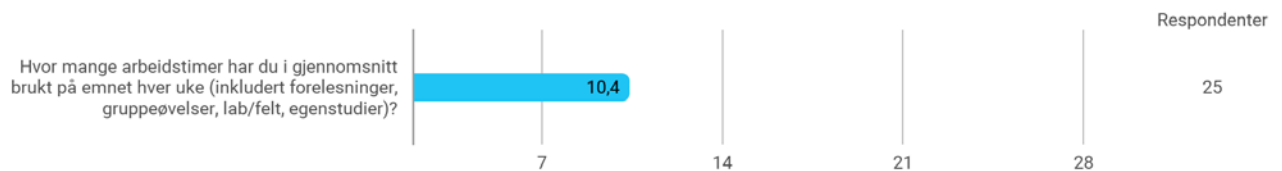
En stor gruppe med variert bakgrunn: informatikk, HVL, kognitivt vitenskap, og enkelte studenter med avansert programmeringskunnskap. Det er vanskelig å finne et nivå og en form som passer alle. Det store auditoriet i ViilVite egner seg dårlig for regulære forelesninger - hovedsaklig grunnet mangel på en stor tavle. Selv om man setter inn en tavle, er avstanden til denne for stor.

Første obligatorisk oppgave ble for omfattende og vanskelig. Dette ble påpekt i flere kommentarer og blir rettet neste gang. Kun ca. 20% studenter gjennomførte evalueringen, så tilbakemeldingene gir ingen troverdig oversikt. En konstruktiv del av tilbakemeldinger angående forelesninger klaget på at det ble for lite koding og at kodeeksempler ikke ble kjørt når de ble vist. Dette kan forbedres neste gang.

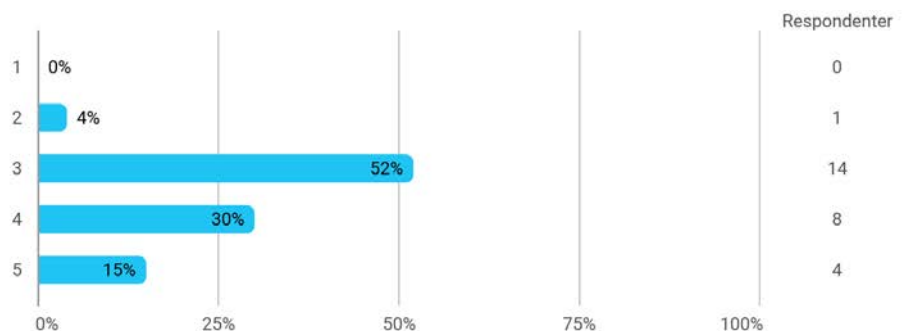
Er du?



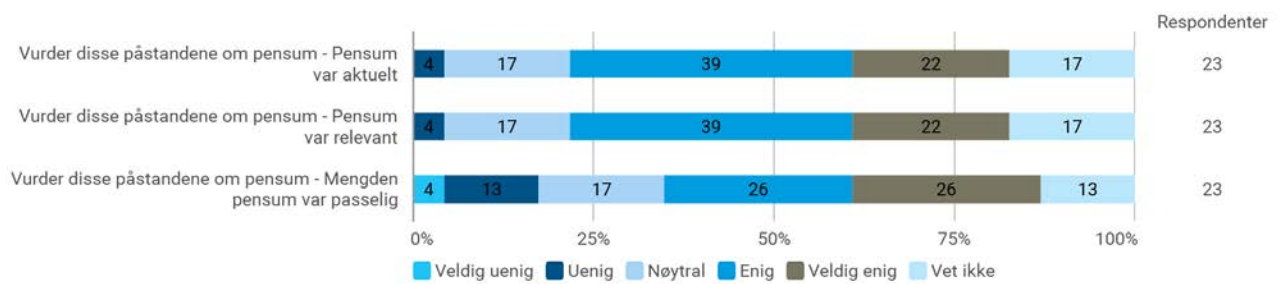
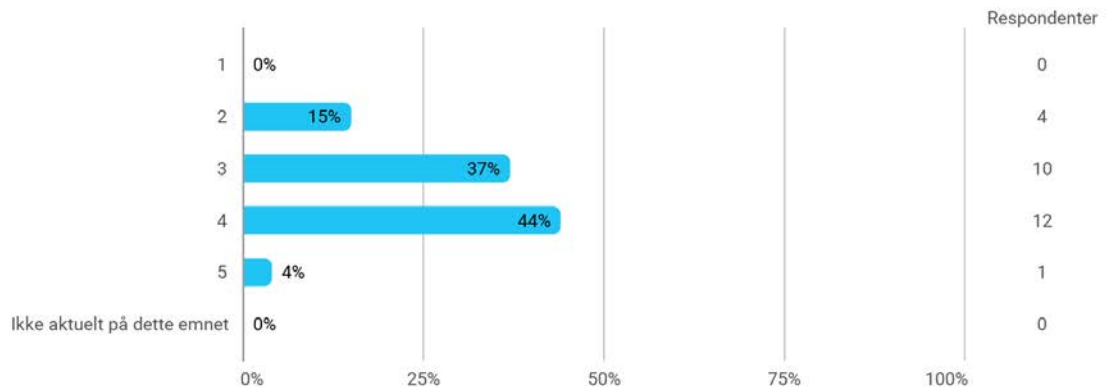
Er du? - Annet



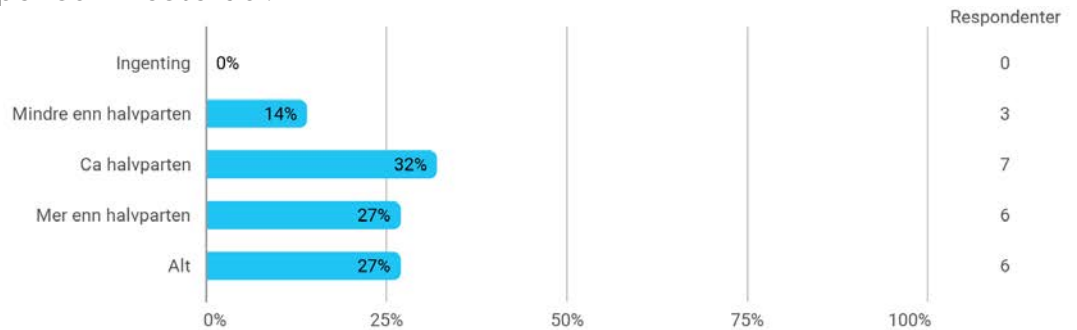
Hvor mye teoretisk kunnskap har du tilegnet deg på dette emnet? (1 = ingen, 5 = mye)

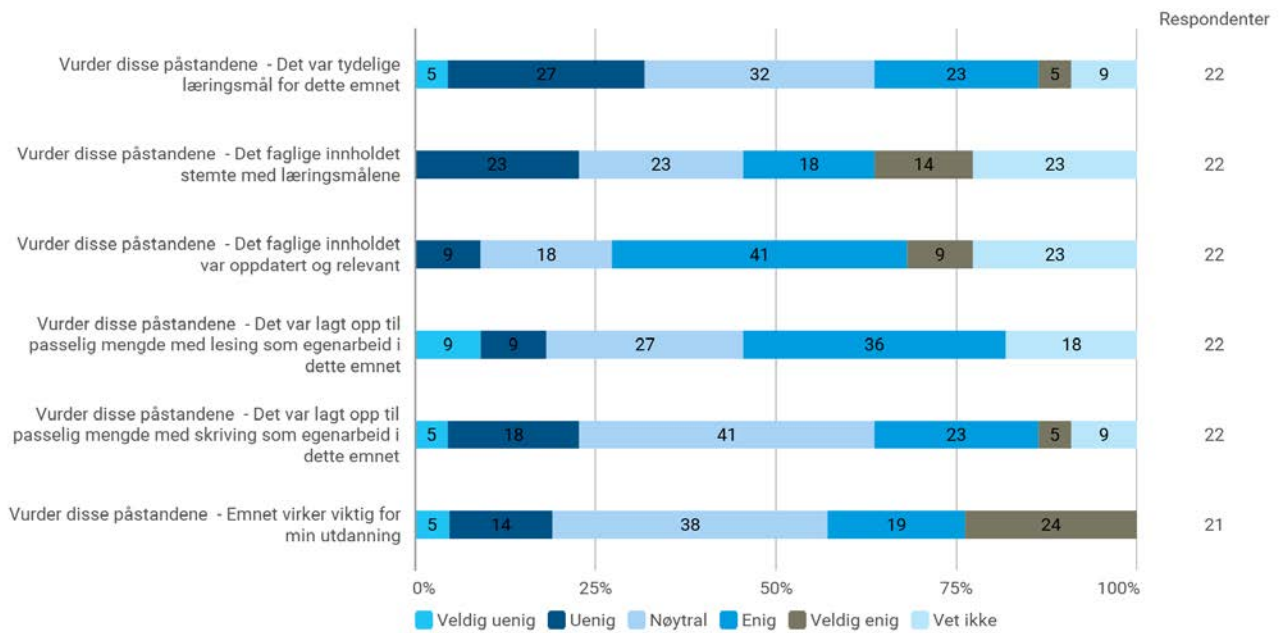


Hvor mye praktisk kunnskap har du tilegnet deg på dette emnet? (1 = ingen, 5 = mye)

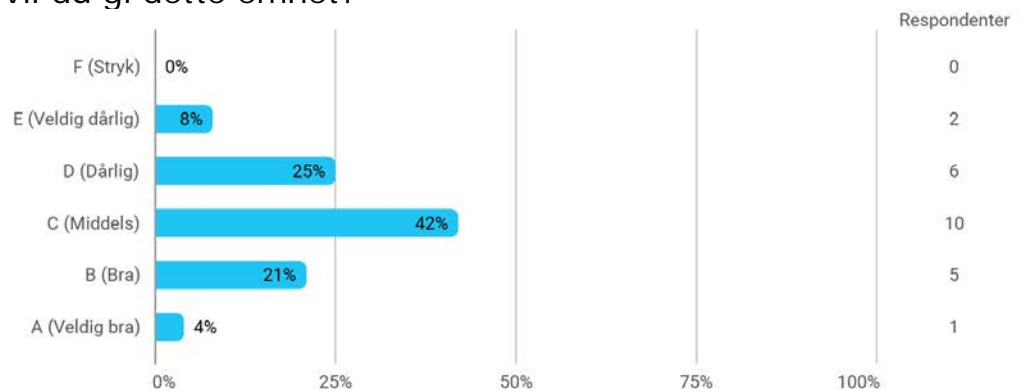


Hvor mye av pensum leste du?





Hvilken karakter vil du gi dette emnet?



Hva likte du mest med dette emnet?

- Flinke undervisningsassistenter som gjorde det mulig å komme seg gjennom faget. De burde ha vært forelesere isteden for den faktiske foreleseren.
- Veldig interessant å lære en helt ny måte å programmere på.
- Kunnskapsrik foreleser og aktuelt emne
- Å lære en ny måte å programmere på
- Hvor kort koden kan bli
- Foreleser var grei
- Var spennende å lære et funksjonelt programmeringsspråk, absolutt et språk jeg kommer til å bruke videre.
- Gruppeundervisningene! Studentassistentene var utrolig flinke og gikk gjennom pensum på en god og pedagogisk måte.
- God lærebok
- Fikk lære om funksjonell programmering
- Jeg likte å lære meg funksjonell programmering, og dermed å øke forståelsen for ulike programmeringsspråk og -paradigmer.
- gruppetimene
- Å lære å programmere i Haskell, da jeg kom over kneiken
- å lære å kode Haskell, bli utfordret til å tenke programmering på en ny måte, i forskjell til mer standar objektorientert programmering.

- Haskell
- Innsatsen fra gruppelederne og den andre obligatoriske oppgaven. Disse har gjort forståelsen for/i faget bedre.
- Noe nytt å lære
- Oblig 2, å programmere programmer som gir valg til en bruker
- Lære et nytt programmeringsspråk, og gruppelederne Ragnhild og Knut Anders.
- Introduksjon til funksjonelle språk.

Hva likte du minst med dette emnet?

- Foreleseren:
Han hadde dårlig opplegg i timene. Man lærte lite av undervisningen hans og de få gangene han faktisk prøvde å skrive en kode selv prøvde han ikke å kjøre den. Få tall av gangene var koden riktig og det kom ikke tydelig frem hvordan den koden han prøvde å skrive skulle brukes.
- Vanskelighetsgraden på første oblig.
Ikke fasit på ukesoppgaver og obliger om man av ulike grunner ikke kan gå i gruppetimer (jeg har feks deltidsjobb med faste vakter som gjorde at jeg aldri hadde mulighet til å gå).
- Første oblig. Spesielt på grunn av feil i eksemplene som gjorde det veldig vanskelig å forstå oppgavene.
- første oblig. Språket var alt for komplisert og uttydelig hva som skulle gjøres. Det var flere, inkludert meg selv, som brukte like lang tid på å forstå oppgaven som å løse den.

andre oblig langt, langt bedre skrevet.
- Synes vanskelighetsgraden på første innlevering var for høy. Ville heller foretrukket å ha én ekstra innlevering, der første er bygd opp slik at man skal lære seg (eller bli testet på) det grunnleggende i funksjonell programmering/haskell.
- Noen eksempler var vanskeligere å forstå i forelesning enn ved å lese boken
- Emnet er bra, men foreleseren er passe ubrukkelig. Både forelesningsnotater og forelesninger var dårlig, han gikk igjennom mye når ingen hang med. Og var dårlig på å forklare på alle måter. Den første obligatoriske oppgaven var helt på tryne formulert, helt uforståelig hva man skal gjøre uten hjelp til å tolke oppgaven. Den neste var bra, men det var fordi den var skrevet av gruppeleder og ikke foreleser, igjen foreleseren som er problemet.
- Forelesningene
- Vanskelig å forstå innholdet på powerpointene
- Jeg likte ikke det voldsomme spranget i vanskelighetsgrad fra ukesoppgavene til første oblig. Den var altfor vanskelig i forhold til det vi hadde lært fram til da.

Jeg liker heller ikke at tidligere eksamener ser ut til å forutsette at vi kan algoritmer som ikke er pensum i dette faget. Med det tenker jeg spesielt på grafteori og spesifikke algoritmer som DFS som var nødvendige på eksamen i fjor, men som ikke er en del av pensum til dette faget.

- at de ikke ble vist så mange kodeeksempler i forelesing
- At innleveringene var alt for vanskelige for fort, og at innleveringene stort sett handlet om en ting.
- Første oblig var veldig tøff, og tok veldig mye tid.
Var også til tider vanskelig å forstå sudo koden på forelesningsnotatene, og hadde vært fint mer mer korrekt kode eksempler.
- Obligatorisk oppgave 1
- Den første obligen.
Vanskelig å forstå, demotiverende, og svært krevende resulterte i at denne obligen gikk ut over tiden man hadde til andre fag. Brukte mange titalls timer på denne, uten å sitte igjen med å ha lært noe spesielt. I tillegg følte oppgaven lite relevant, ettersom den var fokusert rundt en svært liten del av det som var gjennomgått hittil.

Siden jeg har hørt at det er gitt lignende klager på oblig fra tidligere år, er det helt merkelig at det ikke er gjort noe med denne.

- Foreleser
- At det finnes deler av pensum som ikke står om i boken men er snakket om på forelesning (og gitt på powerpoint), men ikke så lett å finne info om dette på egenhånd (parsing og hinley-mindler). Lite konkrete eksempler på forelesning, så noe vanskelig å forstå metodene når det blir forklart.
- Vanskelig å lese lysbildene i ettertid, SELV om jeg var på forelesing.
- Når emnet gikk utenfor bøkene måtte man forholde seg til notater som var vanskelig å lese og forstå.

- Oblig 1 var ikke forståelig for 2.års studenter, når selv gruppeledere med A i faget strever i 10 timer for å løse den burde noe revurderes her.
- Latterlig vanskelige obligatoriske innleveringer
- Oblig 1. Den var veldig uforstående da jeg hadde gjort alle ukesoppgavene på forhand. Siste oblig kom også veldig sent og ødla litt for eksamensøvingen da jeg hadde eksamen dagen før innlevering. Dette resulterte med at jeg gjorde det veldig middels på obligene sammenlagt.

Har du forslag til hvordan emnet kan forbedres?

- Mer kjøring av koder i undervisningen. Mer eksempler.
- Kunne vært bedre lagt opp for studenter som ikke kan gå i alt av forelesninger og gruppetimer av ulike grunner.

I tillegg kunne obligene vært jevnere i nivå, eventuelt kunne den lette obligen kommet først (oblig 2 var vesentlig mye lettere enn oblig 1). Jeg tror vanskelighetsgraden på oblig 1 skremte vekk mange studenter og det er vel ikke ønskelig for verken instituttet, foreleser og studenter.

- Få noen til å lese igjennom obligene mht hvor lett det er å forstå oppgavene.
- En ekstra obligatorisk innlevering (på starten) der du blir tvunget til å lære deg det grunnleggende i Haskell
- Animasjon av rekursjon
Legge vekt på at alle typer skrives med stor forbokstav
Instruksjoner for installasjon av Haskell som gir en forståelse for hvordan de forskjellige tingene man installerer fungerer.
- Bytte ut foreleser med en som kan pedagogikk og læring. Og som kan formulere seg på en brukbar måte.
- Første obligatoriske oppgave kan ikke ha som mål å stryke 50% av studentene... Helt idioti å gi en slik omfattende oppgave så tidlig i semesteret.
- Spesielt med tanke på oblig1: Skriv obligene mer kortfattet og forståelige, og senk vanskelighetsgraden betraktelig. Parsing i Haskell er ikke spesielt lett når man er helt ny til språket. Det var mye greiere å forholde seg til formatet på oblig2, så jeg vil foreslå å følge en modell som likner mye mer på oblig2 enn oblig1 framover.

I tillegg: La eksamensoppgavene være relatert til det som er pensum i dette faget, og ikke la det være en forutsetning at man skal kunne pensum fra f.eks. INF102 for å kunne gjøre det bra i INF122.

- mer praktisk kode i undervisningen, dette gjør det lettere å forstå og huske
- Flere obligatoriske innleveringer (4), der de første går på grunnleggende Haskell, og flere mindre oppgaver med varierende vanskelighetsgrad slik at man ikke får en stor vanskelig oppgave som man enten får til eller ikke, men kan jobbe seg gjennom og få til litt og litt
- Første obligene var veldig nyttig, men tok også på kanten mye tid.
Siste oblig var mer bare å jobbe seg gjennom, måtte ikke pushe seg selv til å forstå noe pensum etc. så den var i overkant lett.
Det å ha mer balanserte obliger, og mer tydlige syntaks korrekte eksempler hadde vært en forbedring.
- Forelesningsnotatene burde forbedres
- Den første obligen må endres.
- Gjør enten obligatoriske oppgaver som er innenfor det vi har gjort i ukeoppgaver, eller sett opp ukeoppgaver som gjør oss klar for å gjøre obligatorisk oppgave.

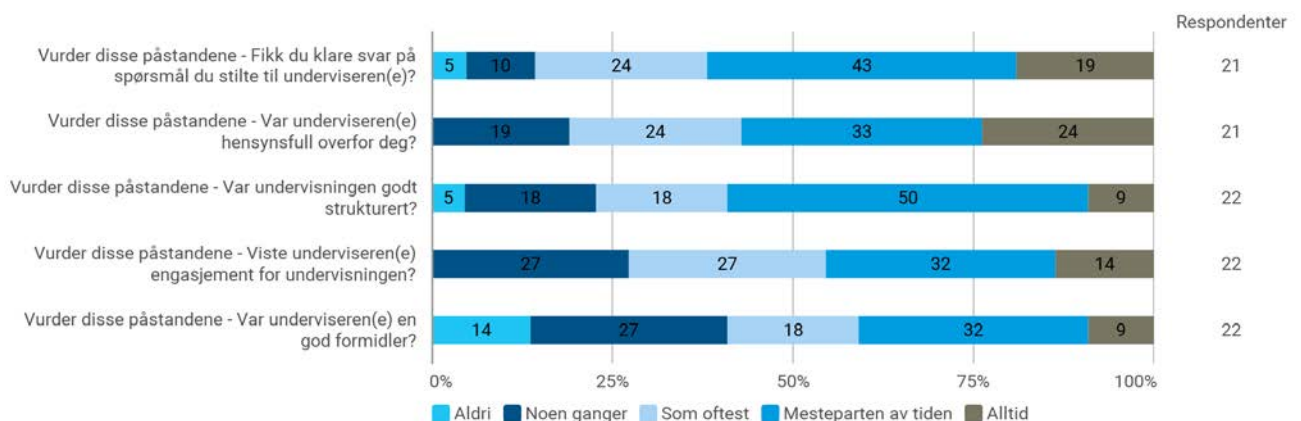
Første obligatorisk var helt forferdelig. Selve oppgaveteksten var så vanskelig formulert at bare å tolke oppgaveteksten i seg selv tok halve tiden vi fikk til råde. Vi driver med programmering, ikke novelleanalyse. Ikke nok med det, selve oppgaven var så vanskelig at man så vidt klarte å fullføre den. Det faktum at jeg og mine venner, SAMMEN, bare klarte halvparten av hele innleveringen sier litt hvor absurd vanskelighetsnivået var. Det var lite læringsutbytte iforhold til innsatsen som ble gjort.

Og til slutt, hvorfor er ikke eksamen digital? Foreleser har ingen gyldig unnskyldning for å ha eksamen på papir. Vi lever i 2018, og PROGRAMMERER. Det eneste som kunne ha vært greit med å ha på papir er trestrukturer, men foreleser liker det ikke, så da er det ingen gyldige grunner for å ha eksamen på papir. **DETTE MÅ RETTES OPP I**

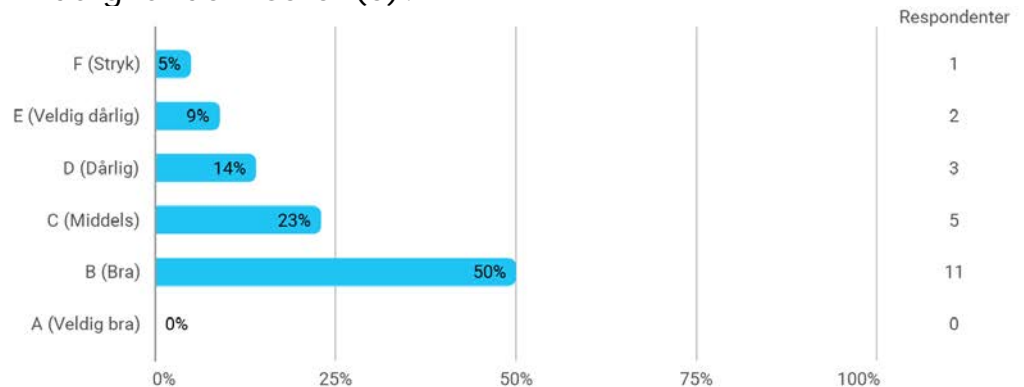
- Hadde hjulper mye om man kunne fått noen konkrete eksempler på f. eks. parsing og typeinferens før obligene.
- Bedre lysbilder. Foreleser burde kanskje være litt mildere i tonen når studenter svarer i forelesningen. Har ikke personlig et problem med det, men jeg vet at flere studenter vegrer seg litt for å svare hvis de blir møtt med en "autoritær" attitude.
- Bruke mer tid til å forklare hvordan man kommer seg fra et sted til et annet i koden og ikke kun fortelle hva man ser
- Mer fornuftig vanskelighetsgrad på oblig 1. Kanskje la gruppeledere lage obligene istedenfor Walicki da han er en veldig god foreleser men forstår kanskje ikke helt nivået på de han foreleser til
- Strukturen på de obligatoriske innleveringene må endres. Fokuser mer på å lære Haskell, og mindre på hvordan å korrekt implementere en parser i språket
- At gruppeledere har viss peiling på hvordan løsningen til obligen er og skal løses da det vart gitt ut forskjellige svar og tips som jeg senere fikk trekk på allikevel.

Tilbakemeldinger på organisert praktisk undervisning:

- Gruppeøvingene var veldig bra og de som gjorde at det var mulig for studentene å ha en sjans til å forstå kurset.
- Veldig fornøyd med gruppene.
- bra at man får en eller to forelesninger på slutten til å gå igjennom deler av pensum valgt av studenter.
- Budskapet til slides kan bli borte når det er så kompakt og så mye
- Gruppelederne var veldig gode, og var mer egnet til å forklare pensum. Ingenting å klage på med de.
- Både krasjkurs og gruppeundervisninger fungerte veldig bra. Det hendte at jeg gikk i timene til både Ragnhild og Knut Anders ila. en uke. Uten dem hadde dette kurset vært fortapt
- Jeg likte at det ble arrangert kræsjkurs i forbindelse med den første obligen, og at det var mulig å få meg seg en gjennomgang av obligene på gruppene etter at de var levert. Det var også nyttig å ha kræsjkurs i forkant av eksamen.
- Veldig flinke gruppeledere som er engasjert og flinke til å formidle kunnskap
- Manglende oversik over gruppetimer og tidspunkter for gruppetimer gjorde det umulig å delta.
- Gruppelederne var flinke
- Gruppeledere gjorde en bra jobb, spesielt med obligatorisk oppgave 2
- Gruppelederne Knut Anders og Ragnhild har hjulpet mye med forståelsen i faget.
- Vil gi en masse skryt til Ragnhild Aalvik og Knut Anders Stokke! (Var ikke på noen andre sine gruppetimer)
De gjorde en fabelaktig jobb med å gå gjennom ukesoppgaver, svare på spørsmål og generelt være støttende.
Beste gruppeledere jeg har hatt så langt i studiet! :)
- Svært gode gruppeledere som var flinke til å formidle stoffet vi skulle lære.
- Gruppeledere gjorde en svært god jobb
- At gruppeledere har viss peiling på hvordan løsningen til obligen er og skal løses da det vart gitt ut forskjellige svar og tips som jeg senere fikk trekk på allikevel.



Hvilken karakter vil du gi underviseren(e)?



Har du forslag til hvordan underviseren(e) kan forbedre sin undervisning?

- Kjør koden du skriver.
Gjennomgå mer eksempler som du kjører.
- Kanskje kode i ghci (ikke bare i en notepad/textedit) så vi kan se hvordan koden kjører og hvordan man kan tyde eventuelle feilmeldinger. I tillegg legge ut koden fra forelesning så vi har noen referansepunkter.
- Være litt mer presis på hva som skal gjennomgås på hver forelesning, og hva som er relevant lesestoff til de forelesningene.
- Iforhold til sist år er han tydeligere, og prater i ett passe tempo.

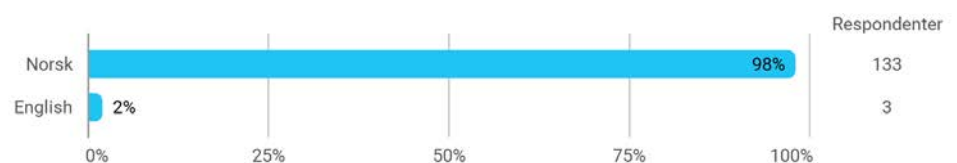
I ett eksempel med quick sort var koden komprimert slik at den ble vanskeligere å forstå.

- Bytte ut foreleser, og beholde gruppeledere
- Kanskje endre powerpointene til å ha et oppsett som likner det i boken.
Vise hvordan funksjonene kan brukes. Det var veldig nyttig i boken for å forstå hvordan funksjoner i funksjonell programmering fungerer.
- Skriv oppgaveteksten til obligene på en mer forståelig og kortfattet måte.

Personlig hadde jeg også foretrukket at funksjonsnavn brukt i eksempler på slides o.l. var på engelsk, siden det kan være litt rart å lære programmering på norsk. For min del hadde det vært lettere å skjønne hva funksjonene skulle gjøre om navnene var engelske, samt ikke forkortet til kun noen få bokstaver.

- flere kodeeksempler
- Bruk verktøy tilgjengelige fra mitt.uib.no for bedre oversikt og enklere deling av informasjon
Lag oversiktlige obligatoriske oppgaver (slik som obligatorisk oppgave 2)
Bruk korrekt føring av eksempler. Dårlig føring gjør det vanskelig å forstå hva som foregår.
- Prøv å være mer forståelsesfull
- Nei
- Litt mindre direkte når han svarer, med tanke på noen studenter som ikke har så god selvtilit/ikke vanligvis er aktive i undervisningen.
- Litt mer strukturerte forelesningsnotater. Ble fort veldig rotete

Språk



Samlet status

