

## A1 Kursusevaluering

Kære studerende

Dine input er vigtige for at afdække hvilke uddannelseselementer, det skal prioriteres at udvikle. Derfor beder vi dig udfylde dette skema. Din besvarelse er anonym, dvs. at ingen underviser vil se, hvilke studerende der har svaret hvad. Svarene anvendes af kursets undervisere, studieleder og undervisningsudvalg på det institut, hvor kurset hører hjemme. Vi vil meget gerne kunne anvende dine svar, så vi håber, at du vil give en seriøs og konstruktiv tilbagemelding!

Du kan besvare evalueringen nedenfor. Hvis der er enkelte spørgsmål, der ikke er relevante, kan du undlade at besvare dem.

341 havde mulighed for at besvare dette skema.

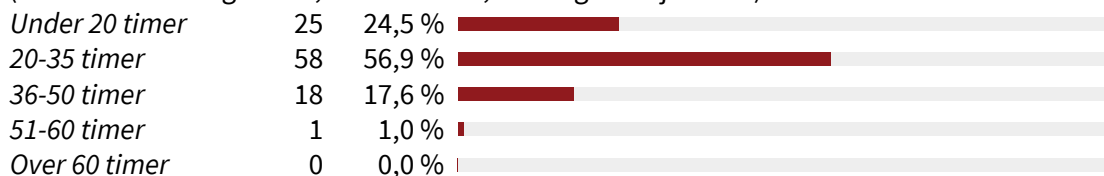
103 har besvaret dette skema.

0 har tilkendegivet ikke at have fulgt kurset.

Svarprocenten er 30,29 % : 103 / (341- 1)

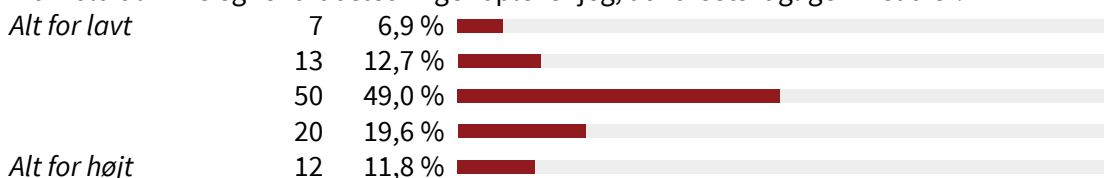
### 1

- 1.1 Min gennemsnitlige ugentlige arbejdsindsats på dette kursus har været på (inkl. undervisningstimer, forberedelse, skriftligt arbejde osv.):

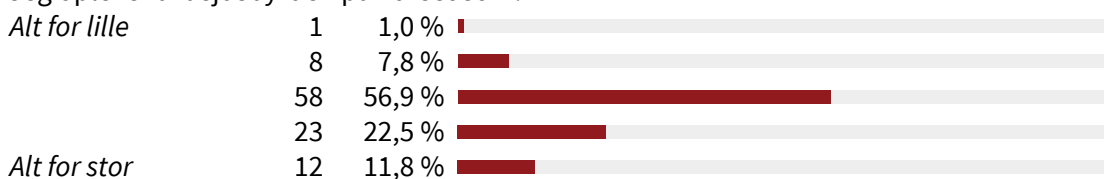


### 2

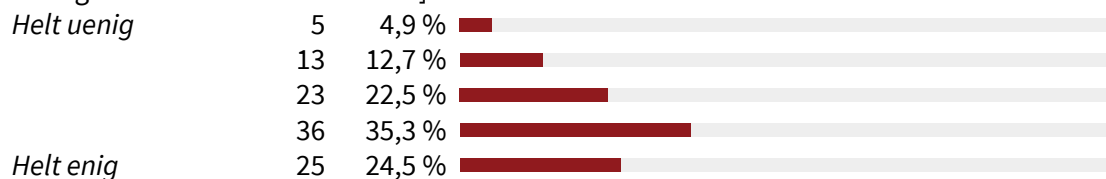
- 2.1 I forhold til mine egne forudsætninger oplever jeg, at kursets faglige niveau er:



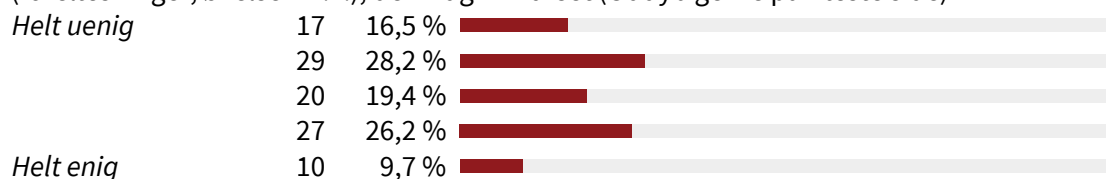
- 2.2 Jeg oplever arbejdsbyrden på kurset som:



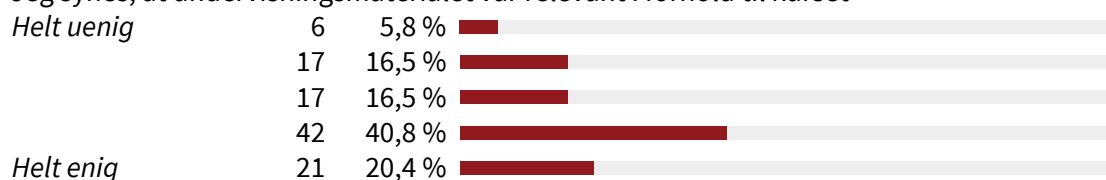
2.3 Jeg mener at have opnået kompetencerne beskrevet i kursusmålene [Se kursets læringsmål fra kursusbeskrivelsen]



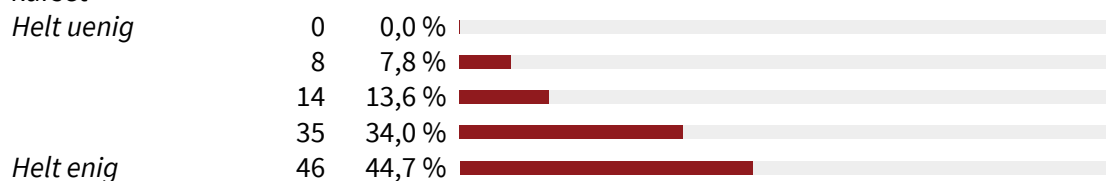
2.4 Jeg oplevede, at der var en god sammenhæng mellem de forskellige delementer (forelæsninger, øvelser m.v.), der indgik i kurset (Uddyb gerne på næste side)



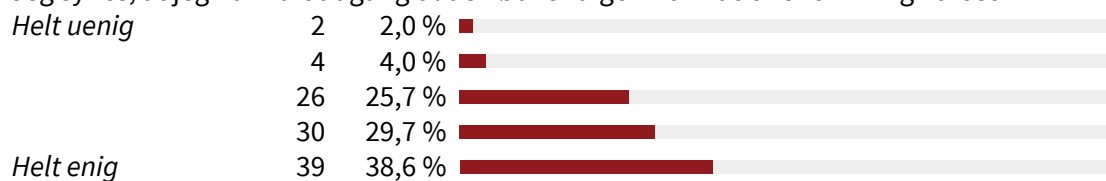
2.5 Jeg synes, at undervisningsmaterialet var relevant i forhold til kurset



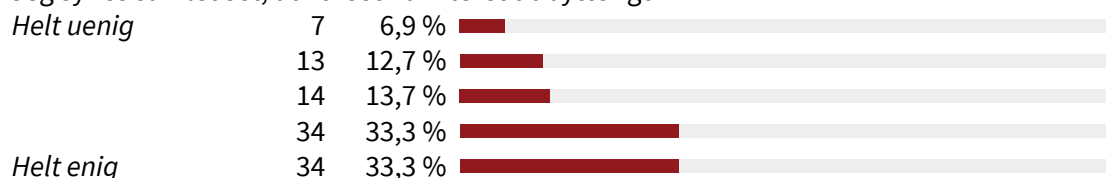
2.6 Jeg synes, at jeg har fået relevant faglig respons på mit skriftlige og mundtlige arbejde på kurset



2.7 Jeg synes, at jeg har haft adgang til de nødvendige informationer omkring kurset



2.8 Jeg synes samlet set, at kurset har været udbytterigt



## A2 Kursusevaluering

Du kan besvare evalueringen nedenfor. Hvis der er enkelte spørgsmål, der ikke er relevante, kan du undlade at besvare dem.

341 havde mulighed for at besvare dette skema.

79 har besvaret dette skema.

0 har tilkendegivet ikke at have fulgt kurset.

Svarprocenten er 23,24 %. : 79 / (341- 1)

### 1

1.1 Hvad fungerede godt ved kurset? Og hvad var grunden?

- *Mængden af øvetimer med instruktør var en stor hjælp. Gode øve opgaver.*
- *Lærebogen har været super god, hvortil forelæsningsne har været et fint supplement. Fedt introduceret til funktionsprogrammering så tidligt, og skønt med fokus på dokumentation (det går gruppearbejdet væsentligt lettere!)*
- *Det var et perfekt kursus.*
- *Det er nogle gode noter, som man kan bruge, når man løser ugeopgaver og afleveringer.*
- *Generelt spændende indhold*
- *Det fungerede godt med øvelsetimerne efter forelæsningsne.*
- *God gennemgang af de basale teknikker som man selv skal lege med senere i opgaver. Forelæserne kunne godt selv se hvad der var lidt svære at forstå hvilket blev forklaret langsommere.*
- *Godt valg af programmeringssprog.*
- *It was nice and challenging.*
- *Jeg fik nok tid at forberede mig og svare på afleveringsopgaverne, men fordi tempoen var ret højt, fik jeg desværre ikke nok tid at lave alle ugeopgaver og der var ingen tid at absorbere materialet, fordi fokuset var på den næste afleveringsopgave.*
- *Det har været fint at lære lidt hvordan man koder, selvom sproget man koder i er ikke noget man bruger på arbejdsmarkedet.*  
*Jeg har lært at kode lidt og det er jeg glad for.*
- *Jeg kunne godt lide, at vi startede med Scratch. Godt at vi kommer igennem så mange forskellige aspekter af programmering (pattern-matching, objekt-orienteret, osv.)*
- *God kultur og introduktion til programmering.*

- *Det er godt der er et kursus der lærer os at programmere.*
- *Ikke ret meget*
- *Mange øvelsestimer, og det at man har ukentlige leveringer som man får tilbakemeldinger på. Lektiecafe.*
- *Både undervisere og vejledere har været rigtig gode til at hjælpe med forståelsen af emnerne.  
Man har fået rigtig god feedback på de afleverede opgaver.*
- *Det større fokus på øvelsestimer. De gav en mulighed for at lære fra en studerende fra senere på studiet, som havde tid til at sætte sig 1-til-1 og forklare koncepter man måske ikke forstod fuldstændigt til selve forelæsningsen.*
- *Jeg synes at torsdagen fungerer rigtig godt, med 2x forelæsning og lang undervisning herefter. Giver god mulighed for at starte på gruppe arbejde eller øvelser*
- *Øvelses opgaver fungerede rigtig godt og hjalp med at forstå afleverings opgaverne uden at være præcis den samme opgave.*
- *God instruktør - siden øvelsesholdet havde samme niveau kunne vi også sagtens gå ud over pensum. F# er et nice sprog.*
- *Det var godt at få kodet en masse og at man konstant skulle kode.*
- *Foruden opgave 7 (markov-chains) synes jeg, at opgaverne formåede at udfordre uden at være uoverkommelige. Jeg opleve stor vækst fra ingen erfaring med f# til en forståelse, som jeg selv er tilfreds med.*
- *Gode emner.*
- *-*
- *Christina er virkelig dygtig.*
- *Instruktorerne var umådeligt behjælpssomme og velinformerede.*
- *Jeg lærte funktionel programmering, og hvordan man laver en rapport til et program.  
F# er god til funktionel programmering, og har nogle interessante koncepter.*
- *Opnåelse af grundlæggende kompetencer indenfor brug af F# samt delvise kompetencer for litteratursøgning.*
- *Øvelsestimer, hvor man kunne arbejde på afleveringer og genafleveringer og få hjælp fra TA.  
Studiecafé om fredagen.*
- *Det var rart at få en a-z i et programmeringssprog, så man også får kendskab til de mere niche egenskaber og features sproget har. Har meget selv lært programmerings erfaring*

*allerede, så kurset var lidt trivielt for mig, men det jo fint nok.*

- *Fint at man havde aflevering i det pensum, man havde haft i ugens løb.*
- *Det fungerede godt, at vi skulle aflevere i samme uge som vi havde om det.*
- *Det var fint, at det ofte var gruppebaseret.*
- *Man kunne læse i bogen og lave nogle af opgaverne, ihvertfald kunne nogle det.*
- *Øvelsestimerne, pga. kompetente instruktører og at have fri i weekenden.*
- *I think that our TA Laura has been wonderful.*
- *Lektiecafé om fredagen.*
- *The curriculum, and most of the assignments seemed relevant.*
- *At få hjælp til opgaverne, både til lektiecafé og af instruktørerne. Grunden er dygtige og forstående instruktører.*
- *Mange gode øvelser.*
- *Afleveringerne var interessante, men havde ofte en fornemmelse af ikke at være klædt på til problemstillingerne.*
- *Øvelsestimerne har været gode, hvor man kunne sidde og arbejde med opgaverne og relativt hurtigt få hjælp.*
- *Gode opgaveoplæg, hvor man har lov til at være kreativ. Der findes ikke kun 1 facit :)*
- *Øvelsestimerne har været rigtig gode. Udbytterige opgaver og god hjælp.*
- *syntes måden vi aflever opgaver på er virkelig godt og feedbacken er god*
- *God intro til programmering i almindelighed.*
- *Ofte nogenlunde sammenhæng mellem øvelser og aflevering.*
- *The abundance of exercises.*
- *Dybdegående forklaringer af syntaks og funktioner i forelæsningstimerne. Det gav dybere forståelse når man senere skulle bruge de lidt mere avanceret værktøjer der var indbygget i fsharp.*
- *Øvelsesvejlederen var rigtig godt til at forklare såvel finde problemer med kode, som ikke fungerede*

- *Fint at bruge et sprog som ingen kendte til*
  - *Undervisningstimerne var super gode. De har ihvertfald været den vigtigste faktor til, at jeg har lært så meget.*
  - *Christina var en rigtig god underviser! Mere af hende! Og langt mindre Martin og Jon - dårlige undervisere.*
  - *øvelserne og opgaverne har generelt set været spændende og sjove at komme igennem*
  - *Meget god fordeling mellem øvelses- og forelæsningstimer. Det er bare fantastisk og noget man skal holde fast i. Det er en uvurderlig hjælp at kunne spørge en TA så meget.*  
*Jeg har lært utroligt meget og det har været super godt, også svært dog, at tempoet har været så højt når man har svage forudsætninger inden.*
  - *Man kunne lave DMA i timerne i stedet for*
  - *Man kom meget i dybden med de forskellige emner, og opklaret mange bagliggende tanker inden for programmering. Det var også gode første opgaver til at komme i gang med at lære at programmere, især hvis man ikke havde gjort det før.*
  - *The instructors were great. The proportion of lectures to practice-time was also well-chosen. Getting an introduction to the different programming-paradigms was a good idea.*
  - *Forelæserne, især Jon Sparring. Der bliver ofte brugt humor, hvilket er godt.*
  - *PoP har generelt for mig været et meget svært fag at takle. Skriver en længere mail i stedet.*
  - *Jeg føler jeg har fået godt styr på F#, men føler mest det skyldes at have siddet selv og googlet og prøvet sig frem med små udfordringer.*
  - *Øvelsestimerne var gode, det var godt at der var så god mulighed for at få hjælp af instruktorerne.*
  - *Stor ros til instruktorerne for deres utrættelige kamp for at svare på alle vores alt for mange spørgsmål.*
  - *At lære et nyt programmeringssprog. Og gøre det grundigt.*
- 1.2 Jeg vil foreslå følgende forbedringer (OBS! Kommentarer vedr. individuelle undervisere angives på evalueringsskemaet for den pågældende underviser)
- *At aflevere en aflevering fredag/lørdag når man får sidste undervisning til opgaven lørdag er ikke ideelt, jeg vil foreslå at køre afleveringerne forskudt en uge ligesom i DMA*
  - *Måske kunne der gøres lidt mere ud af formuleringen af afleveringsopgaverne. Enkelte gange har det virket til, at opgavestiller har haft en bestemt type løsning i tankerne, men det har ikke været explicit. F.eks. 7g, løste vi med RegEx, selvom en løsning vha. List og pattern-matching nok havde givet mere mening i forhold til kursusforløbet.*

- ...
- *Det virker ikke til, at der er særlig meget struktur over kurset. Afleveringerne er til tider alt for svære, og de er helt umulige at løse uden virkelig meget hjælp. Det er derfor frustrerende, at man ikke kan lave en aflevering. Det kan godt være, at man får længere tid til at lave den, men det hjælper ikke noget, hvis der fx alligevel er ferie, og derfor ingen hjælp er at hente. Derudover er det utrolig træls, når man tager i lektiecafe om fredagen, og ikke kan nå at få hjælp, fordi man enten er gået ind i et forkert lokale, selvom vi har fået at vide, at der er lektiecafe i lokalet, eller at der simpelthen bare er for mange studerende, der skal have hjælp i forhold til hvor mange lektiehjælpere, der er. Jeg savner, at der er en sammenhæng mellem forelæsningerne og de øvelser/afleveringer, som vi skal lave i løbet af kurset.. Der bliver brugt tid på alt for mange basale ting, og for lidt tid på at gennemgå hvordan man egentligt programmerer, og hvordan koder hænger sammen.. Det virker også utjekket, at der ofte kommer rettelser ud til afleveringerne på en mail. Jeg vil anbefale, at der også kommer bedre styr på holdinddelingerne fra starten, så mange ikke skal rykke hold efter en del uger. Det er også forvirrende, at der er så stor forskel på, hvad der egentligt forventes af en til afleveringerne*
- *Læg emnet "Fejl og undtagelser, input/output, internet" efter objektorienteret programmering; meget af indholdet i kapitel 19 benytter sig af indbyggede klasser i F#, og det er ekstremt svært at forstå uden at have arbejdet lidt med klasser først*
- *Generelt var torsdagsforelæsninger langt mere brugbare end tirsdagsforelæsningsne. Derfor var der relativt meget opgaveløsnings tid, der var ineffektivt i løbet af ugen, og alt for lidt tid til overs efter torsdagsforelæsningsne. Jeg ville have foretrukket at den lange forelæsning lå om tirsdagen.*
- *Måske mere opgave relevante forelæsninger da jeg af og til føler at man ikke nødvendigvis kan bruge så meget af forelæsningsne til arbejdet, på den måde at man bagefter tænker at man ikke ville have kommet bagud overhovedet hvis man ikke kom.*
- *Nogle af opgaverne var dårligt defineret og "problemløsning" delen var for lille.*
- *It felt rather unstructured at some points.*
- *Det ville være godt, hvis der var bedre sammenhæng mellem det der blev gennemgået til forelæsningsne og det som øvelsesopgaverne omhandlede.*  
*Det ville være rart, hvis afleveringsopgaverne var formuleret mere præcist, da øvelseslærerne ofte rettede dem meget firkantet.*
- *Der var alt for mange afleveringsopgaver, og ikke nok tid at bare lave øvelserne og absorbere materialet bedre. Det vil sige, at min fokus har været at bestå alle afleveringsopgaver og ikke så meget at besvare alle ugeopgaver og tjekke at jeg har forstået alt. Jeg vil foreslå afleveringsopgaver hver måned, med resten af tiden til øvelser og spørgsmål. Vi fik heller ikke svarene til alle ugeopgaver, så vi kunne slet ikke tjekke at vores svar var korrekt. Der var kun nok tid at spørge instruktørerne om en eller to opgaver men ikke om alle. Måske kan det også forbedres, så at studerende kan bare få sendt svarene efter øvelsestimerne.*
- *Der har været mange problemer med kurset og nogle af mine medstuderende og jeg har været meget frustreret med selve kurset og meget utilfreds.*

1. Bogen er ikke speciel god til afleveringer og den er skrevet på den måde at man skal havde kodet før. Det kunne være lækkert med et lille introduktion til hvordan man skal læse den og forstå de forskellige bokse der er i hele bogen.
2. Instruktorerne retter på meget forskellige måder og de er ikke enige hvordan man laver opgaverne. Da har det derfor været svært at få hjælp af en instruktør og få rettet ens aflevering af en anden.
3. Der er har været mange problemer med selve afleverne;
  - (a) Afleveringerne tager rigtig lang tid at lave nogle gange og det er irriterende at vi bruger omkring +15 timer på hver aflevering.
  - (b) Der er har været ret mange fejl i mange af afleveringer og dette må i gerne dobbelt tjekke inden i udgiver afleveringen. Det har været problematisk at havde kæmpet i flere timer på en afleveringen for så at få at vide der er en fejl eller en opgave er for svær til vi kan lave den. Samtidig har jeg fået en genaflevering på en af mine afleveringer, fordi der var en fejl i selve problemformuleringen i opgaven, så jeg havde lavet den rigtig, men skulle genaflevere fordi selve opgave formuleringen var forkert.
  - (c) Jeg ved godt at kurset hedder "(..) problem løsning", men mange af opgaverne er formuleret så dårligt at det er svært at finde ud af hvad man overhovedet skal gøre.
  - (d) Det har været svært at finde ud af hvor meget ens kode skal kunne, da nogle gange skal det være perfekt og andre gange er det okay hvis man har givet et forsøg.
  - (e) Jeg (og andre) føler at vi bliver nød til at lære os selv at kode, da afleveringerne ikke rigtig har haft noget at gøre med pensum, eller det vi har skulle læst i bogen. Dette har også gjort til at bogen har været meget upraktisk.
- 4) Det kan godt være lidt hårdt at vi får en meget svær aflevering for, når vi har ferie, fordi vi "har mere tid", men dette er ikke sandt og det er mere presset når den f.eks. ligger oven i juleferien, hvor man har planer med familien og behænte næsten alle juledage.
- Sørg for at læse opgaverne ordentligt igennem inden de gives. Når det er meningen at vi selv skal definere ting i en tvetydig opgave er det fedt at få at vide så snart vi får opgaven.
- Undlade at bruge eksempler som indeholder emner som kursisterne ikke er blevet undervist i endnu.
- - 1) Det vil være rart at der bliver taget højde for, at bare fordi vi har ferie, betyder det ikke vi har mere tid. De to store afleveringer (uge 7 og uge 11) har været for omfattende og vi havde flere gange problemer med at få hjælp til et helt konkret problem.
  - 2) Mere fokus på at forsøge at komme med en løsning, i stedet for at denne løsning skal køre perfekt, kunne være rart. Så vil man have det bedre med at aflevere en kode, der ikke nødvendigvis var perfekt. Nævn det evt kort til forelæsninger eller i opgaven.
  - 3) Bogen er et godt opslagsværk til de små generelle ting, men der er flere større ting, jeg savner i bogen (2D-Arrays, mere kompliceret eksempel på patternmatching og evt. i en rækkefølge, som også passer til afleveringerne)
  - 4) Når afleveringerne rettes har vi i min gruppe ofte fået vidt forskellige feedback, selvom vi nogle gange har haft de samme fejl. Det kunne være rart med en mere ensformig feedback. Det har også gjort, når man har fået hjælp af en instruktør, men er blevet rettet af en anden, så har man fået fejl i afleveringen, selvom instruktøren sagde det var fint.
  - 5) Afleveringerne har været primært fokus, og øvelserne er kun sjældent blevet lavet da afleveringerne har taget overhånd. Hvis i ønsker øvelserne skal blive lavet, kan i overveje at lave lidt færre af dem, der der ofte var virkelig mange - især hvis afleveringerne skal være så store. Ellers kort lidt ned på afleveringerne, så man ikke behøver at bruge tiden ti øvelserne



*på dem, i frygt af man ellers ikke når dem.*

- *Bogen.  
Ingen grund til at programmere i F#.  
Sæt fokus på hvordan man programmerer, før der sættes så meget tid på problemløsning...*
- *Øvelsesbesvarelser skal lægges ud, ellers ser man ikke sine fejl. DatØk skal ihvertfald have diskret matematik, ellers bliver afleveringerne ALT for svære. Kurset egner sig til nogle der har programmeret før. Ikke til nogle der er nye*
- *Øvelsesoppgaverne behøver en kraftig overhaling, og bør skrives helt om. Tilbakemelding om hvordan dette bør gøres er gitt en egen mail og møte ila kurset. Som minimum bør de ha en sammenheng med afleveringsoppgaverne, og de bør ha en stigende vanskelighetsgrad. Det er en total mangel på forklaring av helt basics i F#, og øvelsesoppgaver som gjør at man blir god på det.*

*Kun de som har programmert tidligere forstår F#.*

*Det er uforståelig hvorfor F# brukes i dette kursuset - dette bør tas opp til diskusjon med flere fagfolk! Det er ingen andre universitetet (og selvsagt ingen virksomheter) som bruker dette språket.*

*Flere intstruktører på lektiecafe.*

*Instruktørene i øvelsestimene bør få bedre instruksjoner.*

*Fasit/løsningsforslag til øvelsesoppgaverne bør gis.*

- *Ingen foreslag.*
- *Mere sammenhengighet mellom forelæserne, da det godt kunne virke som om der var interne uenigheter om, hvordan noget burde undervises på. Dette kan være en sund tilgang senere på kurset, når de studerende har en baggrund og mulighed for at bedømme hver forelæseres mening, men til et introduktionskursus kunne det bare være frustrerende ikke at være sikker på, hvilke approach der ansås som værende "best practice".*
- *Mere hjælp til at komme i gang i starten. Kan være svært for personer der ikke har erfaring at komme i gang med Editors, terminal osv.*
- *I burde overveje at lave opgave 7 om. Selvom vi fik ferien (hvor man ikke kunne få hjælp) til den også, var den for lang og for svær!*
- *Bedre opgaver og mere udspecificerede. Der er for stor forskel i hvor svære opgaverne er. Læs opgaverne igennem før I udgiver dem.*

*Læg en mere detaljeret plan ud over hvad vi gennemgår til forelæsningsne - eventuelt med slides. Der er stor forskel på hvor relevante de er for folk med erfaring, og det kunne være fedt at vide om det giver noget at dukke op.*

*Vælg enten f# og undervis i funktionel, c# og undervis i objekt-orienteret eller skift sprog undervejs.*

- *Jeg observerede en vedvarende miskommunikation og dårlig koordination mellem de*

*forskellige undervisere.*

*Bogen som forelæseren selv har skrevet var helt ufatteligt dårligt skrevet og ikke specielt god at læse. Faktisk læser ingen i den bog - jeg forsøgte, men her til sidst lader det til at være bedre blot at side og forsøge sig frem.*

- *Jeg mener, at opgave 7 var for svær. Den kan eventuelt skæres lidt ned eller placeres senere i forløbet.*

*Jeg kunne godt have tænkt mig én enkelt opgave, som ikke kræver, at man har tilegnet sig nye kompetencer, men i stedet fokuserer på at få finpudset tidligere emner. Eksempelvis ville jeg gerne have arbejdet mere med exeptions. Denne opgave kunne placeres lige før forløbet med objekt-orienteret programmering.*

*Jeg synes godt, at man kunne have undladt Scratch opgaven. Den var sjov at lave, men nu ærgrer jeg mig over at være gået glib af en opgave med større udfordring og læringspotentiale.*

- *Måske lidt lavt niveau.*

- *Bedre opgave beskrivelse.*

- *-*

- *For ofte fejl i opgaveformuleringer*

- *Maybe consider more bigger projects, perhaps as excercises.*

*Really do consider using a better book, the book did not contain good information about the subject, most students had search for it elsewhere. If you want to write your own book, maybe consult with the students -*

- *Christina kan måske stå for kurset fremover?*

- *Der var ofte lidt til ingen sammenhæng mellem forelæsninger og afleveringsopgaver, og dette gjorde afleveringsprocessen hektisk og besværlig. Dette blev mere og mere aktuelt, som kurset skred frem.*

- *1. Der var rigtigt, rigtigt ofte fejl i aflevering og annonceringer på Absalon. Det er ikke sjovt at skulle lave sine afleveringer om, blot fordi man har lavet dem i god tid.*

*2. Jons bog indeholder ikke selv korrekte kodestandarder, og variabelnavngivning sætter ikke et godt eksempel. Det gør desuden koden sværere at læse.*

*3. ImgUtil er dårligt lavet, og følger ikke god kodestil. Winforms virker ikke på Mac for mange - så det skal nok fjernes fra pensum*

*4. Vores instruktør var/er mega langsom til at svare vores opgaver*

*5. Objektorienteret programmering er virkeligt dårligt i F#. Ja, man kan lave et program der bruger objekter og klasser, men man kan ikke lave et program der KUN bruger objekter og klasser. Hvorfor snakke om paradigmer, når vi ikke kan følge dem?*

*6. Tidsafstanden fra man lærer noget i en forelæsning til at du skal bruge den i en aflevering*

er for kort. Vi kan kun lave vores afleveringer i god tid fordi vi er gode til at programmere i forvejen.

- Jeg synes ikke forelæsningsne gav noget som man ikke kunne læse sig til i bogen, så jeg holdte hurtigt op med at deltage.  
Derudover synes jeg at der burde offentliggøres løsningsforslag til øvelsesopgaverne, således at man kunne kigge på det til inspiration.
- Jeg har haft meget svært ved problemløsning delen. Det bør forbedres ved følgende:
  - 1) Yderligere litteratur vedr. problemløsning.
  - 2) Udlevering af eksempler på løsninger til øvelsesopgaverne.
  - 3) Der er stor forskel på instruktørens niveau. Min instruktør, som er andenårsstuderende, faglige niveau i F# har været lavt i forhold til andre instruktører, som har gået længere på studiet.
  - 4) Det er svært at nå øvelsesopgaverne. Derfor fik jeg oftest lavet 1 eller 2 øvelsesopgaver og gik herefter til afleveringen. Derfor bør være en bedre sammenhæng mellem øvelsesopgaver og afleveringsopgaver.
  - 5) En revurdering af øvelses 7i, da den var meget svær.
  - 6) Læringsstilen på kickstartkurset var godt. Den kan muligvis inspirere.
- Evt. forelæsning senere på dagen. Jeg oplevede at meget få mødte op til forelæsningsne.  
  
Desuden kan opgavebeskrivelserne i afleveringerne godt formuleres meget mere specifikt i starten af kurset. Nogle gange kunne det lidt føles, som at lære et nyt sprog, mens alle opgavebeskrivelserne stod på fremmedsproget. Specielt når man ikke har programmeret før.
- Syntes virkelig der mangler en særskilt uge der fokuserer på debugging. Jeg kan allerede finde fejlene i mit program, men har hjulpet mange medstuderende som simpelthen ikke har nogen ide om hvor de skal starte og hvad deres error messages betyder. Og ha mere fokus på problemløsningsdelen, altså hvordan omsætter man tanke til kode helt praktisk, og uden at få en milliard compile errors.
- Jo flere eksempler jo bedre
- Flere eksempler på færdige programmer/funktioner
- Det var for svært for helt nye. At programmere kan være virkelig svært at begynde på, og det kræver at man har grundforståelsen. Dette var der nogle få som fangede, mens resten ikke rigtig forstod noget i store perioder (mig selv inklusiv).  
Det gør at man forfalder til at blive 'inspireret' af andre som forstår det, fordi tidspresset er for stort, og læringen for komprimeret.  
Hvis man vil øge læringen, skulle man derfor evne at tage de svage elever i hånden, så de også fik grundforståelsen på plads. Det vil give dem mod, og mulighed, for at prøve sig mere frem.  
Samtidig synes jeg ikke der var nogle sammenhæng mellem forelæsningsne, øvelser og opgaven. Jeg synes det var ærgerligt at øvelsestimen var anlagt på, at der kun var mulighed for at få hjælp og ingen reel undervisning.
- Jeres inddeling af grupper fungerede ikke i alle tilfælde. Ikke alle gruppe-medlemmer forberedte sig til lektionerne og ikke alle gruppe-medlemmer bidrog til arbejdet. Det viste sig især ved de gruppe-opgaver, der har en arbejdsbyrde svarende til en gruppe og man storset stod med det hele selv.

- *Bedre undervisningsmateriale, bedre gennemtjekket opgave, da mange af dem føles som om de har været hurtigt støt sammen eller har indeholdt dårlige beskrivelser eller fejl, som ender med at have betydning for opgaven.*

*Kort sagt bedre undervisningsmateriale, bedre organiseret og gennemtjekket opgaver og bedre formulering af opgaven i opgavebeskrivelsen med mere konkret information.*

- *I think that, when the course is 15 ECTS, the workload is ridiculous. I also think that many of the assignment were worded misleading which lead to unnecessary confusion and time wasted. Furthermore, i think it is disrespectful of the time of the students to wait until last minute to notify about the amount of assignments. At last i think that there is a problem, not rooted in the course, but at the faculty about respecting other people. I think there is a strong need for some political correctness, to say the least. In my exercises, people were making fun of one of the other TA's for being non-binary. This symptom of narrow-mindedness and arrogance towards people outside the norm is something i have noticed alot in the hallways, and at the social gatherings at the faculty. It is a hostile study environment and not acceptable.*
- *Der virker ikke til at være en stor sammenhæng mellem læsning, forelæsninger og afleveringer. Emnet er måske det samme, men de redskaber man får præsenteret i bogen og til forelæsninger, gør os ikke i stand til at lave afleveringen, og det har været meget frustrerende. Samtidig er afleveringerne ofte dårligt formulerede og kan give misforståelser.*
- *In the functional part of the course, there was a large disconnection between what was taught at the lectures, and what we had to do in the exercises and assignments.*
- *At underviserne sætter sig ind i, hvad ugens aflevering handler om og taler sammen om, hvad de hver især underviser i, så der er en rød tråd.*
- *For studerende uden erfaring med programmering, er opgaverne på dette fag meget svære. Læringskurven er så stejl, at uden hjælp fra både erfarne holdkammerater og hjælpsomme instruktører ville dette kursus være umuligt.*
- *Det er godt med meget klare beskrivelser af hvordan opgaverne skal løses i starten. Dette har dog været svingende i afleveringsopgaverne*
- *Der bør være færre fejl i opgaverne og mere udførlige opgave beskrivelser. Jeg har brugt alt for lang tid på af gætte mig frem hvad der menes.*
- *Bedre sammenhæng mellem materialet gennemgået i forelæsning, og afleveringernes indhold. En del fejl i både bog, og opgaveformuleringer, samt manglende svar i bog på konkrete opgaver i oplæg.*
- *Sammenhæng mellem undervisning, øvelser og afleveringer har haltet i blok 2.*
- *der vare flere tilfælde i midten af kurset, hvor forelæsningerne anvendte relativt simple eksempler, hvorimod afleveringerne var mere komplicerede. Hvilket betød, at man ikke fik helt så meget ud af forlæsningen.*
- *Nogle opgaver var ret voldsomme, og man kunne måske ændre lidt så de forskellige opgaver kom til at være lidt mere lige i forhold til det arbejde de kræver.*

- *Nogle gange hang øvelser og afleveringer slet ikke sammen. Afleveringer var ofte svære at tolke, og forstå hensigten med hvad der forventes i opgaverne.*
- *A clearer and more concise structure, if not vision, for the course and its role in the students professional competency.*
- *Tydeligere rege ang. hvad vi må, og ikke må i visse stadier af studiet. Eksempelvis fik vi på et tidspunkt at vide vi ikke måtte bruge mutable variabler til funktionelt programmering. Hvilket giver mening i sig selv. Men da vi så overgik til Imperativt programmering, troede jeg at reglen stadig var i spil. Indtil min instruktør oplyste mig om at reglen kun var til en specifik uge-aflevering.*
- *Der har ikke været lige så stor sammenhæng mellem læsematerialet og forelæsninger som der var i DMA, synes jeg.*
- *Andet/mere læringsmateriale*
- *Mere sammenhæng mellem POP forelæsningerne og arbejdsopgaverne. Ofte er der slet ingen sammenhæng, og derfor har jeg oplevet mange der har undladt at komme til forelæsningerne*
- *opgaverne har op til flere gange indeholdt fejl, som blev rettet senere og i nogle af de tidligere programmer hvor de blev rettet af et script, var formuleringen af opgaven lidt uspecifik sådan at man fik lavet noget forkert på grund af en misforståelse af opgaven. generelt set kunne opgaveformuleringerne godt sikres at være læst igennem en ekstra gang, da man fik fornemmelsen af at de ikke var læst igennem.*

*personligt synes jeg at nogle af forelæsningerne blev meget af det samme hvor man gik igennem de samme ting op til flere gange, jeg kan godt se nødvendigheden for dette for folk der ikke har programmeret før, men det kunne være lækkert med måske en hurtigere gennemgang el. lign. for folk der allerede har programmeret en del før. og som derfor hurtigere kan sætte sig ind i tingene og nok også har set dem før.*
- *Opgaverne kunne godt være lidt forvirrende nogle gange*
- *Synes at forelæsningen var meget kedelige. Synes det ville have været bedre at lære objektivt før funktionel.*
- *Jeg savner rigtig meget meta omkring alt ting. At tingene bliver sat ind i kontekst og man bliver hjulpet lidt igang.*

*Især i starten var problemet ikke at forstå/løse selve opgaven, men mere at visualisere opgaven og forstå hvordan man får noget fra papir over på pc.*

*Det har været rigtig svært at komme ind i tankengangen med hvordan computeren processerer alting og hvordan det skal stå. Hvis man måske kunne komme med flere eksempler og mere blød snak. Hvor bruges det, hvordan kunne en opgave se ud osv.*
- *Det blev ret forvirrende og ret "rodet" på et tidspunkt gennem forløbet, føltes det som om. Forelæsninger, øvelses opgaver samt afleverings opgaver virkede til at være 3 forskellige ting, igennem nogle af forløbene. Mange af arbejdsdelene var også "forkerte" og skabte forvirring.*

- *I come from a different field of studies and have never in my life even attempted to program before. From that perspective, the learning curve is incredibly steep. The first few weeks were alright, but there was too little time to actually practice and internalize techniques. In consequence, although I was reasonable at the problem solving-part, I could not have done any of the late assignments without (possibly too) extensive help. Although I will probably (just) pass this course, I do not feel in any way confident in the technical programming aspect. It seems doable given practice, but not in the time available. More instructors would also have been helpful, since it became clear early on that a lot of time would be spent waiting for help when reaching a point where one is stuck. Drop the "Scratch"-intro; Since there is so much to learn in such a short time, it seems like an unnecessary diversion (or rather, its usefulness is not in proportion to the very precious time wasted). Occasionally, the tasks given and the subject matter presented seemed slightly out of sync, especially when having spent hours on solving a problem which gets explained in a lecture the day after. I didn't bother looking at the practice-tasks after week 4 or 5, since there wasn't time for it anyway. All in all, I was prepared for a challenge considering it is a renowned institute, but even so, I would suggest to prospective students to prepare for a lot of hard work with little sense of achievement. Even when finishing an assignment after much hard work, I never got the feeling that I would have ever managed it without a lot of help and patience from instructors or other students. Again, this is my subjective experience, starting from absolute zero. I should also reiterate that the actual problems to be solved within a program seemed doable for the most part, but any attempts got frustrated by my lack of actual programming skill, which in turn made trying out and improving the original solutions to the given problems all but impossible.*
- *Brug Python til at introducere programmering. Det bruges på mange andre universiteter (f.eks.: [wiki.python.org/moin/SchoolsUsingPython](http://wiki.python.org/moin/SchoolsUsingPython)) og er meget mere tilgængeligt for folk, som aldrig før har beskæftiget sig med programmering. Man lærer andre programmeringssprog senere hen i datalogi-uddannelsen; det bør være allervigtigst i Pop at overhovedet kunne skrive programmer, forstå hvad en datamat er, forstå hvordan et program er opbygget og oversættes til maskinkode, hvordan man laver klasser, metoder, felter osv. At bruge Python ville gøre denne grundlæggende forståelse af programudvikling og programstruktur meget mere tilgængelig for folk, som enten i et beskedent omfang eller slet ikke har beskæftiget sig med programudvikling.*

*Specifikke paradigmer som funktionel programmering kan komme senere. At gennemtvinge F# fra start er en unødvendigt høj og excentrisk standard. Det elitære argument med, at man vil undervise en "mere rigtig" måde at skrive programmer på fra starten ved at bruge F# holder ikke. Indlæringskurven gøres kun højere, fordi man både skal forstå, hvordan man skriver et program, samt forstå et helt baggrundskoncept for at kunne tilfredsstille nogle krav til ens kode. At starte med Python ville dog gøre, at indlæringskurven er lettere. Man ville først lære at skrive programmer på enkelt og let maner, og bagefter ville f.eks. det funktionelle paradigme blive introduceret ved at kræve, at koden i afleveringen skal overholde visse krav eller benytte bestemte strukturer; f.eks. afleveringen, hvor der skal skrives rekursive funktioner.*

*At forstå programmeringsparadigmer er meget nemmere, når man allerede har en forudsætning for at kunne forstå dem. Med andre ord: det er nemmere at forstå forskellige måder at skrive programmer på, når man i forvejen kan finde ud af et skrive programmer. Almindelige mennesker går ikke rundt og ved, hvordan man skriver programmer i et obskurt sprog eller ved, hvad det funktionelle programmeringsparadigme er.*

*Det andet argument med, at F# er et multiparadigmesprog og dermed bedst egnet til undervisning, holder heller ikke; Python er også et multiparadigmesprog og understøtter*

*funktionel, imperativ og objektorienteret programudvikling. Python er langt bedre egnet til undervisning end F# er, det er nemmere at læse og det er et meget nyttigt redskab ude i den virkelige verden - modsat F#, som absolut ingen nytte har uden for kurset. Hvorfor spille de studerendes tid med et sprog, som de lykkeligt og lettet vil glemme alt om, så snart Pop er overstået? Hvorfor ikke gøre dem en tjeneste og hjælpe dem i deres fremtidige karrierer ved at lære dem et ekstremt velunderstøttet sprog, som de nemt kan benytte i både private og arbejdsrelaterede ærinder?*

*Hvis man starter et hvilket som helst kursus i programudvikling andre steder end Diku, er det ekstremt sandsynligt, at man starter med et enkelt sprog som Python. Og så må man spørge sig selv: er det alle andre universiteter og kurser, der har taget fejl mht. valg af programmeringssprog som introduktion til programmering, eller er det en selv?*

*At bruge Python ville desuden harmonere bedre med Dma, idet dette sprogs syntaks minder meget om den, som man bruger til at skrive algoritmer i pseudokode i Dma. Af denne grund ville det også være nemmere at teste algoritmer ved at skrive dem i Python; man ville altså kunne integrere Dma og Pop meget mere, end tilfældet er lige nu. Der er mange flere ressourcer tilgængelige for Python, og sprogets understøttelse på diverse platforme er i fuldkommen top. Sproget har et imponerende stort økosystem og fællesskab. Men F#? Bare for få dage siden måtte Jon Sparring oplyse, at 13i i Pop ikke blev til noget grundet problemer med Winforms. Flere gange i andre afleveringer har der også været problemer med F# på Mac og Windows. Og jeg tænker bare... hvad foregår der her? Er man virkelig så fastsat på dette uduelige sprog, selv når dets mangelfulde understøttelse bevises gang på gang - foruden alle de andre grunde, som jeg har opridset?*

*Der er ingen god grund til at bruge F#. Brug Python i stedet for alle fremtidige Diku-studerendes skyld.*

*Jeg kan forstå, at det muligvis er en trist tanke at skulle kassere F#-bogen, som er blevet udarbejdet, men denne frygt er heller ikke nogen god grund til at holde fast i F#; i så fald ville hele ræsonnementet om at fortsætte med F# hvile i et følelsesladet eller ideologisk argument. Man kunne sagtens skabe en lignende bog om Python, som er lavet specifikt til Pop på Diku; f.eks. kunne man gøre ligesom Kbr-bogen i Dma, hvor udvalgte afsnit fra et større værk udplukkes. Derudover kan de mere teoretiske afsnit fra den eksisterende F#-bog bibeholdes, da de ikke handler om selve F#-sproget.*

*Denne side giver inspiration til nogle bøger, man evt. kan arbejde ud fra:  
[realpython.com/best-python-books/](http://realpython.com/best-python-books/)*

- Læs ovenstående.
- Jeg vil foreslå at huske at give lange beskrivende navne til variabler, da dette vil gøre kodeeksempler meget nemmere at følge med i og forstå. Når det bare er x, y, z, og forkortelser på 3-4 bogstaver, giver man ret hurtigt op med at prøve at forstå.
- Der er behov for bedre sammenhæng mellem forelæsninger og opgaver. Det er nødvendigt med en bro imellem teori og opgaver i form af konkrete eksempler eller andet mere håndgribeligt. Teorien fra forelæsningerne er alt for svær at overføre til afleveringerne, og det gør at den bliver ubrugelig i sidste ende. En meget vigtig pointe er, at der glemmes at mange studerende ikke har programmeret før. Man kan følge med i de første par afleveringer, men så stikker niveauet af og mange bliver hægtet af og mister modet. Der mangler eksempler og hjælp for som kurset er nu, er mange studerende frustrerede og demotiverede fordi de ikke kan løse de opgaver de bliver stillet - fordi de ikke får kompetencerne til det. Man kan ikke spørge en person der ikke kan dividere, hvad 10 divideret med 2 er, før man har lært dem om division og hvordan man gør. Det er sådan det har føltes at have pop - frustrerende og demotiverende.

- Opgavestilingen i faget bærer præg af at være venstrehåndsarbejde. Det var ikke ualmindeligt at der kom rettelser til afleveringsopgaverne 1-2 dage før deadline. Det kan naturligvis forekomme at der opdages uventede fejl i nye opgaver, men mange af rettelserne var af en karakter som burde kunne opdages og rettes allerede inden kursusstart. En del af opgaverne er heller ikke nye, og fejlene i dem bør derfor være velkendte. Som eksempel kan nævnes opgave 6, hvor det bliver kommenteret at en af funktionerne let kommer i en uendelig løkke, så man skal introducere en separat måde at stoppe den på. En hurtig google-søgning afslørede at en studerende med et dansklydende navn havde siddet med præcist det samme problem i præcist den samme opgave for to år siden. Det kan altså ikke være noget ukendt problem. Det kan være fair nok at beholde en sådan fejlkilde, da folk skal lære at håndtere den; men hvis man finder det nødvendigt at rundsende en mail for at gøre opmærksom på det, bør man også kunne finde ud af at skrive det i opgaveteksten til at starte med – og ikke dagen inden afleveringsfristen. Det er respektløst over for de studerende at udvise så lidt engagement i stilling af eksamensopgaver.

Som nævnt ovenfor er eksempler en vigtig del af læringsprocessen, og forelæsninger er særligt velegnede til denne del. Det bør der blive gjort mere ud af (som sagt, med tilbundsgående forklaringer af de enkelte trin).

En indikator for hvor godt man klarer det er antallet af spørgsmål der bliver stillet.

Studerende stiller spørgsmål når der er noget de gerne vil have forklaret en ekstra gang eller hvis der er et videre perspektiv i emnet. Hvis de ikke stiller spørgsmål kan det derfor have to årsager: Du har forklaret emnet så grundigt at der ikke er nogen tvivl og at alle har forstået alt til bunds – det kommer ikke til at ske, for man kan altid stille spørgsmål til næste niveau; og en studerende der har forstået et emne er en entusiastisk studerende der gerne vil vide mere. Mere sandsynligt er det derfor, at en mangel på spørgsmål er afstedkommet af at folk har forstået så lidt at de ikke ved hvilke spørgsmål de skal stille – de er opgivende og tror ikke på at forelæseren kan forklare tingene på en måde de kan forstå, og mener derfor ikke at det er besværet værd at prøve at spørge. Mange vil sikkert sidde med spørgsmålet ”Kan vi gå to forelæsninger tilbage og starte helt forfra? Jeg har ikke forstået noget af dette her.”; men det er ikke et spørgsmål de fleste studerende har lyst til at stille, så de tier.

Jeg hæfter mig i denne sammenhæng ved at kurset startede ud med masser af spørgsmål til forelæsningsne, hvilket svandt ind til næsten ingenting – blot for at eksplodere igen i Christinas forelæsninger. En forelæsning hvor der ikke bliver stillet spørgsmål bør derfor få alle alarmklokker til at bimle, for det betyder sandsynligvis at dine tilhørere ikke har forstået hvad du har sagt.

Jeg har oplevet at der ikke er tilstrækkeligt med instruktørressourcer til rådighed, om end jeg ikke ved om dette gør sig gældende på alle hold eller kun på de hold jeg har været på. Jeg oplevede at instruktorerne (Mikkel og Emilie) kæmpede en brav kamp for at få svaret på alle spørgsmål og få givet folk den hjælp de skulle bruge, men man måtte ofte vente en halv time eller mere på at få hjælp fordi der var rigtigt mange der skulle bruge rigtigt meget hjælp. Som en studerende uden forudgående programmeringserfaring var jeg selv en af dem der skulle bruge relativt meget hjælp, idet jeg ofte ikke selv kunne finde fejlene i mine programmer (hvis du er ved at lære kinesisk nytter det ikke meget at fejlmeddelelserne selv er på kinesisk, hvis ikke du har nogen der kan hjælpe med at tyde det for dig). Med nogle af afleveringsopgaverne oplevede jeg at jeg fik lavet 2/3 af opgaven i øvelsestimerne, men halvdelen af tiden sad jeg og ventede på hjælp. Havde jeg kunnet få hjælp når jeg havde brug for det, kunne jeg sagtens have lavet hele opgaven inde for øvelsestimerne og have haft tid til overs til øvelsesopgaverne.

Jeg vil tilråde en kraftig opmænding på instruktørfronten eller en omprioritering af hvor instruktorerne er. På holdene med de studerende der allerede kan programmere (hvoraf mange i forvejen ikke dukker op til undervisningen da de ikke har behov for hjælp) er det måske fuldt ud tilstrækkeligt med en instruktør til 20-30 studerende, mens det på holdene uden programmeringserfaring ikke ville være ublu med en instruktør pr. 2-3 studerende. Den første uge med scratch er spild af tid. Man bruger en uge på at lære det, hvorefter man smider det ud og starter forfra fra bunden af med F#. Der er lidt til intet af dem man når at lære med scratch der kan overføres til F#, så det tjener ikke noget læringsmæssigt formål. Scratch kunne godt bruges til at lære nogle basale ting om loops og branching som godt ville



kunne overføres, men det kræver at man har tid til at sætte sig ind i det og få en forståelse der er dybere end lige hvordan scratch har implementeret det; og det når man ikke på én uge. Det ville sandsynligvis kræve 2-3 uger før det var umagen værd.

Jeg ville i stedet anbefale at man startede med tekstbaseret, dedikeret nybegyndersprog såsom Microsoft Small Basic. For totale nybegyndere gør det ikke den store forskel om man starter med et tekstbaseret eller grafisk sprog, da alt alligevel er nyt, og erfarne programmører bliver bare forvirredeovre skiftet til et grafisk interface. Small Basic er relativt intuitivt at gå til og der medfølger en god og pædagogisk manual. Det er ikke et stærkt typet sprog og ville derfor eliminere en af de store startfrustrationer. Det understøtter læring om bl.a. loops, branching, arrays og basal grafik, og man ville derfor kunne bruge det væsentligt længere tid end scratch. Det har en syntaks der minder væsentligt mere om F# end scratch og dét, sammenholdt med at man vil have haft langt bedre tid til at blive fortrolig med de bagvedliggende principper, ville gøre det langt lettere at tage springet til F# end fra scratch, og man ville langt bedre kunne overføre det man havde lært. Der er nogle af de mere avancerede programmeringsparadigmer der ikke er supporteret i Small Basic, så det er stadig nødvendigt at skifte til et andet sprog for at lære de ting, men jeg vil mene at man med fordel ville kunne benytte Small Basic eller noget tilsvarende de første 1½ - 2 måneder af kurset.

Og så lad os da for Turings skyld blive fri for emacs! Der er intet godt ved den editor! Undskyldninger om at "vi skal lære at blive ordentlige teknikere/programmører og skal derfor kunne skrive kode uden hjælp fra editoren" er en gang fis i en hornlygte. Når børn skal lære at cykle starter de på den trehjulede, og siden en cykel med støttehjul – ikke med at blive sat på en full-size mountainbike og sparket ud fra toppen af et bjerg fordi "nu skal de altså lære at blive ordentlige cyklister"! Efter to måneder på studiet er de eneste der stadig bruger emacs dem der aldrig har prøvet andet. Der ingen der skifter til emacs. Lad os da for pokker få lov til at bruge VS Code eller noget tilsvarende fra starten af.

Jeg er heller ikke den store fan af den tvungne brug af LaTeX – i særdeleshed første måneds tvungne brug af den downloadede version i stedet for Overleaf. Jeg brugte 12 timer på at downloade skidt (og det er altså ikke fordi jeg har så dårlig en netforbindelse), bl.a. fordi det skal hente 117 millioner pakker fra en milliard forskellige steder. Efter så endelig at have fået det downloadet, brugte jeg den næste måned på ikke at få lortet til at virke. Det første jeg gjorde efter den tvungne måned var at slette det hele igen – og dermed frigive 6 GB! Det er altså voldsomt meget plads for en teksteditor, når man tænker på at et spil som MS Flight Simulator 2004 kun krævede 2,2 GB – og havde hele verden indbygget!

Det er rimeligt nok at man stiller krav om at opgaver afleveres i pdf-format, men så kan man da i øvrigt være bedøvende ligeglad med hvilket program der har genereret den fil. Man kan lave lige så pæne rapporter i Word (ja, tænk engang!), så lad os dog bruge et program vi er fortrolige med, i stedet for at smide den ekstra komplikation og frustration i hovedet på folk der har rigeligt at se til i forvejen.

Når man starter folk uden programmeringserfaring på datalogiuddannelsen, må man forvente at de kan lave fejl i ting som det ikke er fysisk muligt at lave fejl i; så det vil være fordelagtigt at have noget tech support til dem. Jeg brugte fx ind til efterårsferien på ikke at kunne køre kompilede filer, fordi jeg ikke havde fanget at man skulle skrive mono foran filnavnet når det skulle køres. Hverken min gruppemakker eller to instruktører kunne spotte fejlen.

Man kunne også overveje om man skulle fusionere Programmering og Problemløsning med Diskret Matematik og Algoritmer til ét fag; eller i det mindste sikre en bedre koordinering fagene imellem. Der er et signifikant overlap mellem de emner der bliver dækket i de to fag (om end de bliver tilgået fra forskellige vinkler), og på denne måde ville man kunne skære noget redundans fra og sikre en mere tilbundsående belysning af et givent emne – i stedet for at man lærer om et emne i ét fag i én uge, og så har om det samme emne i det andet fag en måned senere. Dette ville også gøre det nemmere at sikre tilstrækkeligt med instruktørressourcer. Min erfaring er at der er rigeligt med instruktortid i DMA, hvorimod der som sagt er mangel på samme i Pop.

- Drop scratch-delen i starten af kurset.

*Det er mega forstyrrende at der til næsten alle forelæsninger kom 20-30 studerende dryssende for sent i løbet af de første 20 minutter. Det må forelæserne gerne gøre opmærksom på er mega forstyrrende!*

*Afleveringsopgaverne skal være bedre korrekturlæst. Der har været for mange fejl og uklarheder i dem - og da opgaverne i forvejen er tidskrævende, så er uklarheder uholdbare.*

## B Underviserevaluering (Christina Lioma)

Du kan besvare evalueringen nedenfor. Hvis der er enkelte spørgsmål, der ikke er relevante, kan du undlade at besvare dem.

341 havde mulighed for at besvare dette skema.

81 har besvaret dette skema.

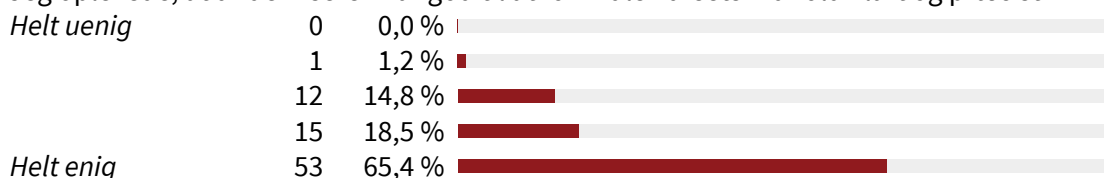
15 har tilkendegivet ikke at have fulgt kurset.

15 har tilkendegivet ikke at have haft denne underviser.

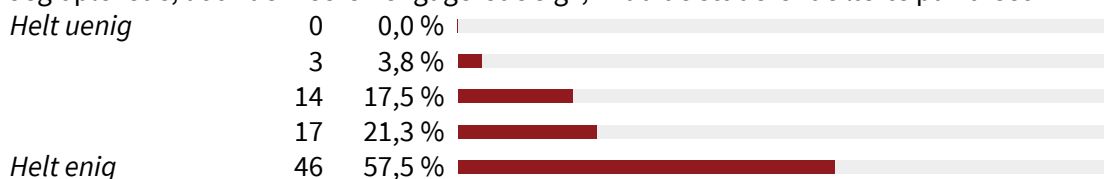
Svarprocenten er 24,92 % : 81 / (341- 1- 15)

### 1

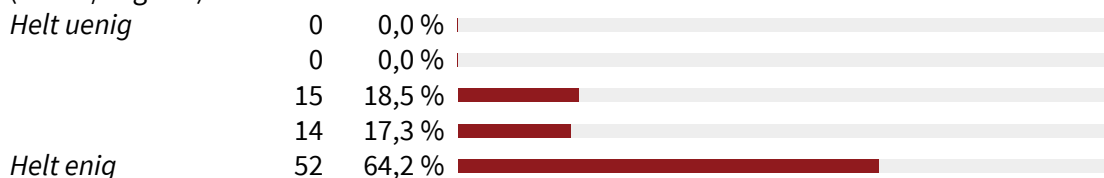
#### 1.1 Jeg oplevede, at underviseren var god til at formidle kursets indhold klart og præcist



#### 1.2 Jeg oplevede, at underviseren engagerede sig i, hvad de studerende lærte på kurset



#### 1.3 Jeg oplevede, at underviseren var god til at udtrykke sig klart på undervisningens sprog (dansk/engelsk)



#### 1.4 Yderligere kommentarer:

- *The same things seemed to be covered several times during a lecture or took long to explain. A little speed up would not hurt:)*
- *Dygtig underviser og gode og relevante forelæsninger*
- *Christina gik lidt mere i dybden med hvordan computeren læser vores kode, og det var meget inspirerende. Jeg kunne bedre forstå mange koncepter og hun var rigtig dygtig med at give eksempler og svare alle spørgsmål.*
- *Den bedste forelæser på dette kursus. Kunne være fedt hvis de andre forelæsere lærte noget af hendes måde at undervise. Hun formår faktisk at forstå hvor meget vi ikke kan.*

- *Sådan skal en forelesning være.*
- *Den bedste forelæser*
- *Lidt forvirrende slides, der af og til modsiger hinanden og hvad hun selv siger.*
- *Hun var den bedste.*
- *I appreciated, that Christina would go through some of the code line by line and make sure, that we understood what every word did and meant.*  
*Not every line should be treated this way of course, but going through a few was really helpfull.*
- *Maybe the pace of lecture was a bit too slow*
- *Mere Christina*
- *Hun er en god underviser.*
- *Er meget koncis og tydelig. Virkelig rart og nemt at følge med i. Niveauet er generelt lidt for lavt for mig, men jeg har også mere erfaring end den gennemsnitlige studerende.*
- *Lidt længere pauser. Lidt bedre til at få forelæsningsne til at passe med tidsgrænser*
- *Hun formåede at tale til alle til forelæsningsen, både dem som ikke forstod så meget, og andre*
- *God og relevant og struktureret undervisning.*
- *God til at forklare og fordybe sig i de detaljer, der var vigtigst for at forstå emnet.*
- *En gang imellem bliver der brugt lidt for lang tid på simple ting, og lidt for kort tid på de mere indviklede.*
- *Christina var fantastisk*
- *S-tier underviser. Godt tempo, tog sig tid til at sørge for alle var med og enormt god til at lære fra sig.*
- *Nogle få af programmeringseksemplerne i bogen og til forelæsningsne kan godt virke lidt for avanceret for nybegyndere der ser det for første gang, synes jeg.*
- *Rigtig god underviser, der ikke blot læste op fra powerpoint.*
- *Virkelig god underviser! God slides, god intuition, god orden. Jon og Martin kan lære meget af hende.*
- *Meget imponerende forelæsningsne , og er lidt ærelig over vi kun havde så få.*

- *Very good lecturer.  
the minute and thorough step-by-step approach is much appreciated, including seemingly obvious matters. In that, clearly the best lecturer.*
- *Jeg tager til forelæsningen for at være passiv.  
Det er dybt ubehageligt at møde op til forelæsninger, hvor forelæseren gerne vil have man er "aktiv"; altså stille spørgsmål og svare på spørgsmål. Det er slet ikke noget forelæsninger som koncept er "gearet" til at kunne.*
- *Christinas forelæsninger var langt de mest udbytterige. Jeg havde en fornemmelse af at forstå mere stof efter én forelæsning med hende end efter flere uger med Jon og Martin. Hun må meget gerne stå for langt mere af undervisningen (helst det hele) eller i det mindste undervise de to andre i hvordan man underviser.*
- *Super forelæser. Meget tydelig og præcis i sit sprog og stor faglig autoritet.*

## B Underviserevaluering (Jon Sparring - Underviser)

Du kan besvare evalueringen nedenfor. Hvis der er enkelte spørgsmål, der ikke er relevante, kan du undlade at besvare dem.

341 havde mulighed for at besvare dette skema.

93 har besvaret dette skema.

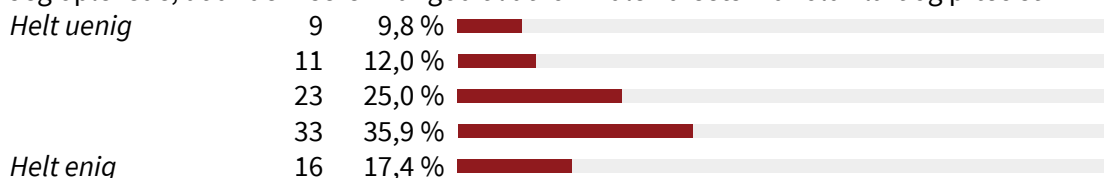
2 har tilkendegivet ikke at have fulgt kurset.

2 har tilkendegivet ikke at have haft denne underviser.

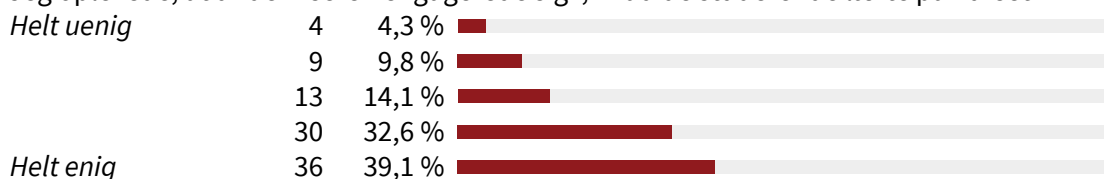
Svarprocenten er 27,51 % : 93 / (341- 1- 2)

### 1

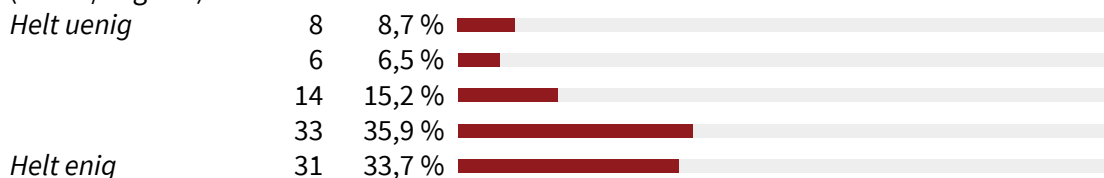
#### 1.1 Jeg oplevede, at underviseren var god til at formidle kursets indhold klart og præcist



#### 1.2 Jeg oplevede, at underviseren engagerede sig i, hvad de studerende lærte på kurset



#### 1.3 Jeg oplevede, at underviseren var god til at udtrykke sig klart på undervisningens sprog (dansk/engelsk)



#### 1.4 Yderligere kommentarer:

- *Pas på at der ikke kommer for mange sidespring i en forelæsning.*
- *Jeg har kun været enkelt forelæsninger. men Jon har virket både klart formuleret, vidende — og sympatisk.*
- *Ved nogle af forelæsningerne gik det for langsomt. Det var heller ikke alle forelæsninger, der virkede lige relevante for de stillede opgaver. Men generelt var det godt.*
- *Sparring har svært ved at gennemskue hvad eleverne egentlig kan, og hvor svært programmering kan være. Han snakker tit til sig selv eller forventer at det er noget vi kan. Gør det grundigere, enklere og tænk på det er noget vi aldrig har lavet før. Ingen grund til at benytte de begreber ingen alligvel kender til.*

- *Han har glemt hvordan det var for ham at starte med at programmere. Afleveringerne er alt for svære, hvilket de stadig er efter 5 rettelser.*
- *Bør ta et kursus i pedagogikk, og en svært nøye gjennomgang av hvordan forelesningene er lagt opp. Det er en ulempe at foreleser og bok har samme forfatter, spesielt når det er såpass lite pedagogisk lagt opp. Spurring har ikke per i dag evnen til å forklare til de som ikke kan det. Jeg opplevde at det var svært at følge med i alle forelesningene.*
- *Jon virkede meget forvirrende og til tider, som om han ikke riktig vidste, hvad det var han underviste i.*
- *Jon er sikkert en super god forsker, men hans forelæsninger er middelmådige og sjældent særligt udfordrende. At læse op i bogen tager 20 minutter, og så behøver man næsten ikke tage til forelæsning. Han er dog altid glad og virker som om at han rigtigt godt kan lide F#, hvilket jo er et plus :)*
- *Undervisningen kunne inspireret af Christinas didaktiske fremgangsmåde. Hun tager udgangspunkt et tema. Giver en kort introduktion. Kommer med et eksempel, som hun bygger videre på. Afslutter med hvad man har lært.*
- *Der ikke helt sammenhæng med bogens (relativt høje) niveau og undervisningens (relativt lave) niveau. Der bliver ofte snakket om ting fra bogen, men "dumbed down"*
- *Din laser var ikke til at se*
- *var nogle gange svært at se laseren når der peges. Bedre ved brug af den tilhørende pind*
- *Jeg synes at niveauet nogle gange blev for højt, og for svært. Samtidig var der absolut ingen sammenhæng mellem forelæsning og øvelser og opgaver, hvilket gjorde interessen markant mindre.*
- *Der blev brugt alt for meget tid på ligegyldige detaljer, og der var ingen overensstemmelse med det, som vi nåede igennem til forelæsninger, og det, som vi skulle lave i afleveringerne.*
- *Hver forelæsning virkede meget ustruktureret. Man følte ikke man kunne bruge det, han sagde. Gentog bare det, der stod i bogen.*
- *Bogen er god, men der er en del fejl der kan ændre betydningen og forståelsen i vigtige afsnit eller emner.*
- *Riktig god og engageret underviser*
- *Dårlig intuition, dårlig struktur, dårlig til at forklare selv basale begreber.*
- *Husk tegnsætning i opgaverne*
- *Helt klart en dygtig programmør, men gennem nogle af forelæsninger følte man det godt kunne gå lidt "langsomt". Ikke at det var trivielt det der blev forklaret, men at der blev lagt fokus på de forkerte ting. Så der ikke var lige så god tid til de mere relevante/forvirrende/svære emner.*

- *Especially in the beginning, and in spite of remarking on the dangers of experienced programmers underestimating the difficulties of n00bs understanding the very basics of programming, I believe he might have underestimated the difficulties in understanding the basics of programming for persons thoroughly ignorant like myself. I will take part of the blame, since it was clearly stated that anything could and should be asked, but still I had a hard time discerning to what degree I had just underestimated the difficulty of the course or to what degree he would just speed through the "obvious". Scratch might in fact have lulled me in a false sense of security in week 1.*
- *Var ret dårlig til at bruge lange variabelnavne, selvom han anbefalede det i hans bog. Dette gjorde kodeeksemplerne svære at følge med i.*
- *Jon er meget teoretisk og fabler løs om sin bog, men der mangler simpelthen konkrete eksempler. Hvis man aldrig har programmeret før, forstår man ikke hvad der foregår. I starten af kurset var teorien endda langt bagved opgaverne - der blev brugt lang tid på gennemgang af binære tal, mens opgavernes sværhedsgrad steg helt vildt.*
- *Jon skal blive meget bedre til at forklare ting i stedet for bare at vise dem. Mange studerende kan ikke se ti linjer kode og straks se hvad de gør - det skal forklares nøgleord for nøgleord når du vil introducere noget nyt.  
Jon gør et hæderligt forsøg på at lave en god forelæsning, men det mislykkes til en vis grad pga. en manglende forståelse for hvor svære og ikke-selvindlysende ting er når man ikke ved det i forvejen.*
- *Jon kan til tider godt stramme lidt op på tempoet. Og så ikke svare på så mange (ligegyldige) spørgsmål fra de studerende.*
- *Nogle gange forklarede han de nemme ting lidt for grundigt - det stod i kapitlet til forelæsningen og derfor virkede det overflødigt at skulle gennemgå igen.*



## B Underviserevaluering (Martin Elsmann)

Du kan besvare evalueringen nedenfor. Hvis der er enkelte spørgsmål, der ikke er relevante, kan du undlade at besvare dem.

341 havde mulighed for at besvare dette skema.

90 har besvaret dette skema.

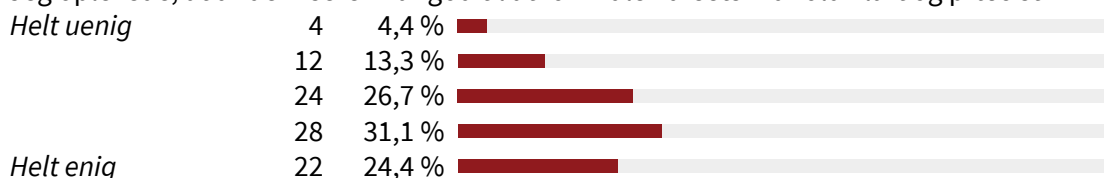
3 har tilkendegivet ikke at have fulgt kurset.

3 har tilkendegivet ikke at have haft denne underviser.

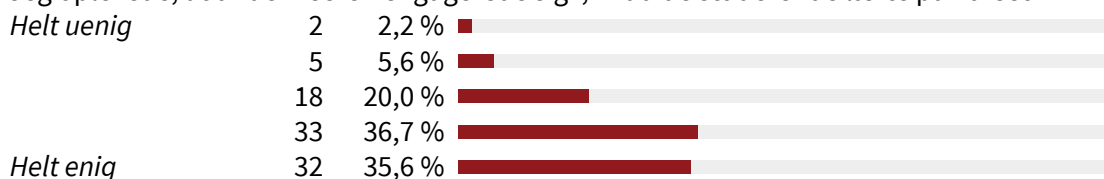
Svarprocenten er 26,71 % : 90 / (341- 1- 3)

### 1

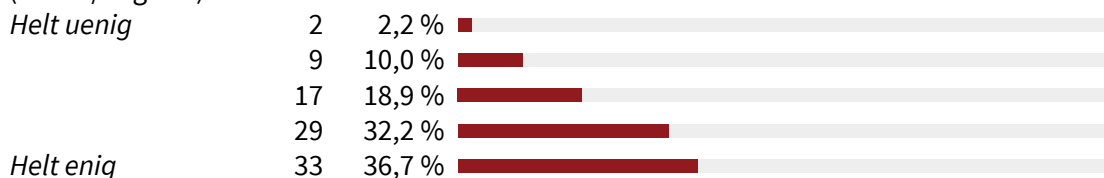
#### 1.1 Jeg oplevede, at underviseren var god til at formidle kursets indhold klart og præcist



#### 1.2 Jeg oplevede, at underviseren engagerede sig i, hvad de studerende lærte på kurset



#### 1.3 Jeg oplevede, at underviseren var god til at udtrykke sig klart på undervisningens sprog (dansk/engelsk)



#### 1.4 Yderligere kommentarer:

- *Din stemmeføring kan blive ret monoton men dit indhold er rigtigt spændende. Det er dog ikke alt du behøver at køre i gennem på terminalen. Jeg tror på at det virker, du kan nøjes med de programmer som tager input fra terminalen.*
- *Forelæsninger/slides har været et fedt supplement til lærebogen.*
- *En engageret underviser, men havde svært ved at ramme de studerendes niveau somme tider.*
- *Utrølig dygtig og engageret, ja. Hans sprog kunne godt blive mere simpelt. Jeg forstod ikke mange af de avancerede ord han brugte, selvom det oftest var yderst simple ting han beskrev.*

- *I forhold til forelæsning på diverse emne, kunne lidt mere intuition godt være med, eller fremvisning af diverse teorier. Forelæsning kan nemt blive meget tung, hvis det kun er teoretisk.*
- *Martin gjorde det okay. Det virker lidt som om at han har beef med Jon? Det skal de vist lige få løst til næste år.*
- *Martin havde ofte nogle lidt mere komplekse kodeeksempler end Jon og brugte ofte mere tid på at forklare dem. Martins timer er mere udfordrende, hvilket jeg synes er positivt.*
- *Undervisningen kunne inspireres af Christinas didaktiske fremgangsmåde.  
Hellere få eksempler, som man gennemgår i dybden end en lang række.*
- *Det blev altså ofte kedeligt og langsomt at følge med i. Til gengæld går det sygt hurtigt når der bliver vist kode og der bliver ikke uddybet særligt. Syntes ofte emnerne kun bliver berørt overfladisk.*
- *Han var meget god*
- *Jeg synes virkelig at det startede forrygende, og gav mig mod på faget.  
Jeg synes dog at niveauet faldt, eller ihvertfald skiftede retning til at blive mere teoretisk, hvilket gjorde interessen markant mindre.*
- *Undervisningen virkede irrelevant. Dårlig formidling - man kan mærke, han er god, men ikke så god til at lære fra sig.*
- *Man vidste ikke helt, hvad man skulle bruge hans eksempler til, da emnet først kom mange uger efter han havde gennemgået det. Læste meget op fra sine powerpoints.*
- *Meget oplæsning fra eksempler på kode. Gennemgang af hvad koden betyder, linje for linje, på dansk, kan godt blive træls.*
- *Forelæsninger kunne være lidt kringlede en gang imellem*
- *Hjemmesiden var god til at se end anden forklaringen af undervisningsmaterialet.*
- *Rigtig god underviser. Nogle gange blev der læst lidt for meget op af hans powerpoint.*
- *Meget uklare slides, dårlig intuition. Mange emner blev slet ikke motiverede - fx hvorfor er det her vigtigt.*
- *Mange gode slides, og okay tempo til forelæsning, men det var i en periode hvor jeg følte mange af tingene ikke hang sammen med det vi lavede uden for forelæsningen. Mange flotte "avancerede" kode eksempler med gennemgang, men føltes også nogle gange at man startede lidt for "højt".*
- *Point one to be taken with a grain of salt, since I had some trouble with the programming itself, so my problems might have started before his lectures. I remember him explaining different algorithms that I sometimes understood and sometimes not quite.  
Good at answering questions and clarifying and clearly very competent. Perhaps sometimes*

*overestimating student's like me (or I was just a slow learner).*

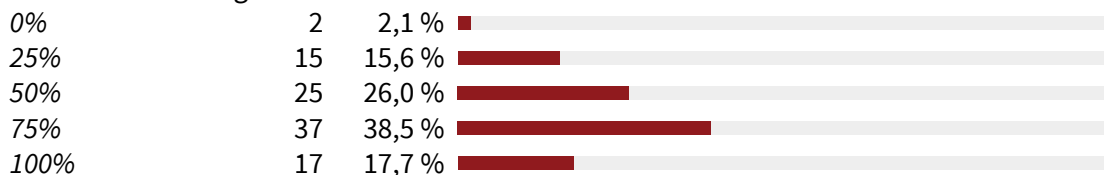
- *Var ret dårlig til at bruge lange variabelnavne, så kodeeksemplerne var svære at følge med i.*
- *For teoretisk og for "internt". Det er som om der glemmes at en stor del af de studerende aldrig har programmeret før, og man bliver tabt hvis man ikke kender begreber på forhånd.*
- *Martins undervisning var meget lig Jons: Et hæderligt forsøg der fejl pga. et for højt abstraktionsniveau. Hans eksempler var lidt mere illustrative end Jons, og han var lidt bedre til at gennemgå dem, men der er langt op til Christinas niveau.*
- *Dygtig forelæser, som er meget faglig engageret og god til at kommunikere sit fag.*

**Evaluering af bog: Learning to program with F#**

341 havde mulighed for at besvare dette skema.

97 har besvaret dette skema.

Svarprocenten er 28,45 % : 97 / 341

**1****1.1 Hvor stor del af bogen har du læst?****1.2 Har bogen været din PRIMÆRE kilde til information om hvordan man programmerer og til sproget F#****1.3 Hvilke ANDRE kilder til information om hvordan man programmerer og til sproget F#, har du haft glæde af?**

- *Microsoft dokumentation i F og C# Slides*
- *<https://fsharpforfunandprofit.com>  
<https://docs.microsoft.com/en-us/dotnet/fsharp/>  
<https://stackoverflow.com/search?q=fsharp>*
- *F# Documentation. Stackoverflow.*
- *Bogen har været nyttig, men det er langt fra alt, man kan finde i den... Ved ikke præcis hvad jeg ellers har brugt, men har googlet en her del*
- *google, slides fra forelæsninger*
- *Jeg har primært søgt information på internettet*
- *Microsofts guide og beskrivelse af alle symboler og operatorer. StackOverflow.*
- *Microsofts dokumentation og andre google sider.*
- *Microsofts egen hjemmeside omkring F#, og tutorialspoint.com*
- *Google, hvis jeg havde et konkret problem.*
- *Andre F# bøger og google.*

- *Søgt meget på nettet fordi man ikke rigtig kunne bruge bogen til afleveringerne.*  
*Har kigget meget i bogen for at finde informationer, i forhold til noget jeg gerne vil kode. Desværre har bogen ikke haft specielt godt information til f.eks. hvordan man ændre en sætning sådan at den står i omvendt rækkefølge og sådan nogle ting. Da har bogen været virkelig dårlig til at kunne hjælpe på nogen måde med afleveringerne.*
- *StackOverflow og diverse andre internetsider*
- *F# documentation fra Microsoft, Stackoverflow*
- *Officiel dokumentation og andre hjemmesider.*
- *Internettet!*
- *Internettet. Spørgesider og Microsoft egen F# sider.*
- *Forskellige sider på nettet. Der er dog ikke stor mulighed for hjælp på nettet, da det er begrænset, hvor meget dette sprog bruges. Det gør det svært, at selvstudier uden bogen, da bogen ikke er god. Derudover er alt ikke forklaret grundigt i bogen. Det kan gøres MEGET bedre.*
- *Stack Overflow*
- *Nettet*
- *Jeg har brukt andre kilder på internett, men skulle virkelig ønske at boken kunne være til mer hjælp. Nærmest all tid jeg har brukt på boken har vært bortkastet, og den burde ikke vært godkjent for bruk som pensum. Den er langt ifra ferdig, og svært lite pedagogisk.*
- *Official F# documentation*
- *Hjemmesiden [tutorialspoint.com/fsharp](http://tutorialspoint.com/fsharp)*
- *Microsofts egen dokumentation, selv om den dog ingen glæde var.*
- *F#'s egne sider på internettet - nemmere at slå det op man skal bruge, end at kigge i bogen*
- *[docs.microsoft.com](http://docs.microsoft.com)*
- *Nettet*
- *stack overflow og microsoft docs*
- *StackOverflow, Microsoft doc. Når man kender til programmering er det tit lettere at slå op ad hoc, end at læse lange kapitler, om noget hvor man ved meget af det i forvejen.*
- *kodestumperne og slides fra undervisningen hjalp en del.*

*Hvis jeg ikke kunne finde mit svar i bogen, søgte jeg på løsninger og tog bare den første side, som omhandlede det emne, jeg arbejdede med.*

- *Internettet*
- *Microsofts documentation på deres hjemmeside.  
Stackoverflow.*
- *Studiekammerater samt diverse forummer på nettet.*
- *Fsharp dokumentationen og tilfældige google-søgninger*
- *microsoft sider*
- *Internettet??*
- *Microsofts egne guides og opslagssider.*
- *Stack-overflow*
- *Microsofts dokumentation*
- *Jeg har brugt microsofts egen dokumentation til slå methods osv op.*
- *<https://www.youtube.com/watch?v=c7eNDJN758U&t=1386s>*
- *<https://fsharp.org/>*
- *<https://www.dotnetperls.com>*
- *<https://stackoverflow.com>*
- *TA's til øvelsestimer & studiecafé. Hvis jeg fik fejlemeldninger, brugte jeg også alt der var at finde på google.*
- *Microsofts dokumentation for F#, i særdeleshed deres "Developer Network" hjemmeside som har god dokumentation for de indbyggede funktioner.  
<https://msdn.microsoft.com/da-dk/visualfsharpdocs/conceptual/microsoft.fsharp.collections-namespace-%5bfsharp%5d>*
- *Online dokumentationen for modulerne.*
- *<https://fsharpforfunandprofit.com>*
- *stackoverflow  
medstuderende  
instruktor  
Microsoft*
- *Medstuderende, instruktor og microsoft*

- *googlesøgninger.*  
*stackoverflow mv*
- *Internettet*
- *Internettet og functional programming using F#*
- *Internettet*
- *Nettet*
- *fsharpforfunandprofit*  
*Microsofts egen dokumentation*  
*Stackoverflow*  
*Diverse andre kodesider*
- *Nettet, rigtig meget Microsofts hjemmeside*
- *Internet kilder så som microsoft*
- *Internettet.*  
*Holdkammeraters hjælp.*  
*Instruktør hjælp.*
- *Bog: Functional programming with F#,*  
*af en eller anden Martin gut.*
- *Diverse sider på Stackoverflow eller microsoft egen kilde.*
- *Instruktør, forelæsere, studiegruppen, og internettets mange sider.*
- *Microsoft dokumentation*
- *Microsofts dokumentation, samt stack overflow*
- *Internettet og andre studerende*
- *internettet*
- *www*
- *Online ressourcer, eks. "F# for fun and profit", "F# cheat-sheet"*
- *Forelæsnings slides*  
*Samt F# hjemmeside*  
*StackOverflow*
- *Dokumentationen.*

- <https://msdn.microsoft.com/en-us/visualfsharpdocs/conceptual/visual-fsharp>
- *Microsoft docs. Faktisk kunne en forbedring være måske at have links i bogen til den rigtige dokumentation*
- *Microsofts documentation.*
- *internettet, herunder stackoverflow og docs.microsoft.com*
- *google  
msdn*
- *Microsofts egen dokumentation, og af og til stackoverflow.com.*
- *Stack Overflow*
- *Microsoft, .NET*
- *F# hjemmeside og stackoverflow*
- *Typisk har der været stackoverflow, microsoft eller diverse youtube videoer. Learning to program with F# er en bog der har en her masse grundviden, men ikke viser hvordan der kan bruges til øvelsesopgaverne eller afleveringsopgaverne. Der har internettet været til meget større brug.*
- *<https://docs.microsoft.com/en-us/dotnet/fsharp/>  
stackoverflow.com  
stackexchange.com*
- *google, især microsoft documentation*
- *Har selv programmeret i flere år, så derfor kom mange ting naturligt, og jeg havde referencer fra andre sprog, til programmering mere generelt.  
Til f# har jeg bl.a. brugt microsofts dokumentation, stack exchange og lign og "f# for fun and profit"*
- *<https://www.tutorialspoint.com/fsharp/index.htm>*
- *fsharp.org  
fsharp fun and profit*
- *Dokumentation på nettet*
- *Mest microsoft docs kilde information til .NET modulet. Ellers diverse andre kilder på internettet.*
- *Stack overflow og diverse andre internetsider - Google.*
- *generelt at google ting*



- *F# for Fun and Profit*  
*MSDNs dokumentation af F#*
- *stackoverflow.com*  
*windows F# hjælpe portal på nettet*
- *Microsofts .NET hjemmeside.*

## 1.4 Har du ofte læst i bogen INDEN den tilhørende forelæsning og/eller øvelse?

Ja	68	70,1 %	<div><div></div></div>
Nej	29	29,9 %	<div><div></div></div>

## 1.5 Har du ofte læst i bogen EFTER den tilhørende forelæsning og/eller øvelse?

Ja	57	58,8 %	<div><div></div></div>
Nej	40	41,2 %	<div><div></div></div>

## 1.6 Har du fundet eksemplerne i bogen nyttige?

Ja	64	68,1 %	<div><div></div></div>
Nej	30	31,9 %	<div><div></div></div>

## 1.7 Hvis du skal nævne NEGATIVE aspekter af bogen, hvad skulle de så være?

- *Jeg oplevede ikke at der var ret mange der forstod indholdet af kapitel 9, det blev meget rodet og jeg fik først ryddet op i dette efter at have været til forelæsning*
- *At eksemplerne i bogen ikke følger F# code formatting guidelines*  
*(<https://docs.microsoft.com/en-us/dotnet/fsharp/style-guide/formatting>)*  
*At bogen ikke findes i trykt form endnu*
- *Jeg mener at der mangler stof til hvorfor man netop skulle vælge at gøre det på den ene måde fremfor den anden. En dybere forklaring ville have været super.*
- *Ikke ret spændende at læse, savner flere eksempler*
- *Den er svær at finde rundt i. Stikordsregisteret er højst mangelfuldt. Der er en håndfuld slåfejl/sætningskonstruktionsfejl, som bør rettes ved en grundig korrekturlæsning. I forhold til kurset har det virket som om de stillede opgaver ikke er blevet formuleret med udgangspunkt i bogen.*
- *Jeg synes til en vis grad at den er mangelfuld på visse områder i den forstand at jeg ser teknikker eller symboler i opgaver som jeg bliver nødt til at google mig frem til hvad det går ud på, efter jeg har ledt i bogen.*
- *Den store baggrunds tekst var meget irriterende.*
- *Det er svært at slå op i den og dermed få inspiration til at løse et problem, hvis man ikke har et stort overblik over programmering.*
- *Bogen er svær at forstå for begyndere og det er svært at finde hjælp på nettet da F# ikke er så populært som andre sprog. Man er tit nød til at finde svar i C# og så prøve at få det til at*

virke med F# syntaks.

- Bogen er skrevet sådan at det er svært at følge rundt i den hvis man aldrig har kodet før. Det virker som om at man skal have kodet før for at kunne forstå bogen. Der gik lang tid før jeg forstod rigtig hvad der indformatonen var i de forskellige dele af bogen.

Som nævnt tidligere er bogen faktisk ret dårlig til afleveringerne. Det eneste jeg (og andre) har brugt bogen til er de forskellige list muligheder. Derfra har bogen ikke været speciel god til at hjælpe os med afleveringerne.

- Den er dårlig som opslagsværk, så hvis jeg gerne vil have svar på enkelte specifikke ting er det ofte nemmere at google.
- Eksemplerne er ofte for små og korte, til det der skal bruges i øvelser/afleveringer. Især i pattern matching - der er kun eksempler, på meget små ting, kunne være rart at se et eksempel på matching over liste.
- Måden at formulere/opskrive de indbyggede funktioner har været forvirrende for dem der ikke er bekendt med den slags notation allerede. Billdeder og kode eksempler ligger også ofte ikke nær den tekst der beskriver eller refererer til den. Det er særligt slemt i object delen. Da dele af teksten er et stort kaos a 'billed' eksempler blandet med normal tekst der taler om noget helt andet. Savner nogle gange lidt mere konkrete eksempler. Særligt på dokumentation af object programmering.
- Føler ikke det er en bog der er skrevet til nybegyndere, selvom det er mening. Mange ting er ikke forklaret helt grundigt, og det er som om det forventes at man ved. Men der sidder faktisk mange der aldrig har hørt om disse begreber eller ved noget om helst om det. Bogen er skrevet i et meget svært sprog, hvilket gør det ikke nemmere at forstå. Lav enkle eksempler og forklar meget mere de eksempler. Bogen er ikke struktureret, kan gøres meget bedre.
- Jeg synes ikke, at den er særlig god til at formidle noget godt. Den er spækket med stavefejl og nogle eksempler er der fejl i.
- Eksemplene, forklaringene, sproket - det meste.
- Den kan til tider have meget lange forklaringer på ting, hvor det er tydeligt at man har prøvet at give en omfattende forklaring så det gav fuldstændig mening, men hvor det i stedet endte med at forvirre mere end noget andet. Det er desværre noget som opstår i de fleste kapitler. Eksemplerne er samtidig lidt for simple til ofte at vise hvordan visse egenskaber ved F# sproget er brugbare.
- Der er nogle gange contradictions i hvad der står (f.eks. brug sigende variabelnavne, når alt altid hedder s, x, osv.). Der er meget overflødig indhold, og nogen gange bliver det vigtige gemt væk af tekniske detaljer. Det er som om at målgruppen ikke helt er afklaret - da jeg lærte at programmere var ressourcerne mindre optagede af detaljerne og det overflødige, hvilket gjorde at jeg hurtigere kunne få en basal forståelse - derefter kan man så meget nemmere forstå detaljerne.
- At den er skrevet helt utroligt dårligt. At forelæseren ikke er i stand til at være kritisk overfor sin bog, og dermed måske tror at den er bedre end den er.

- *Jeg oplevede, at bogen undlod at nævne en række vigtige indbyggede liste, string og array funktioner, som viste sig meget nyttige i løsningen af opgaver.*
- *Jeg kunne godt have haft brug for flere eksempler, eller vedlagt filer til de eksempler i bogen, så jeg selv kunne prøvekøre dem. Evt. flere småopgaver i bogen.*
- *Der er ikke nok information i bogen*
- *Strukturen er virkelig dårlig, tingene bliver forklaret på en kringlet måde og udbyttet er utrolig småt.*
- *Dårlige variabelnavne (brug PascalCase til members og brug mere beskrivene variable!), stavfejl, et stort DRAFT midt på siden*
- *Det er stadig et udkast til en lærebog. Jeg savner en lærer bog ligesom vi har haft på DMA - CLRS.*

*Bogen kunne også linke til relevante kilder, hvor man kan læse mere om, hvordan man eksempelvis kan anvende lister osv.*

- *Der måtte gerne være lidt flere eksempler eller lidt tydeligere, og lidt mindre teori, eller i så fald kunne basis teorien måske samles i starten af hvert kapitel.*
- *Den er måske lidt for vidspændende, den forsøger både at være "lær at programmere F# fra bunden" og samtidigt et opslagsværk som en garvet F#'er skulle bruge undervejs i kodeprocessen. Føler første job udfyldes godt, andet job kan jeg bedre lide at bruge internettet til. Det måske bare personlig preference.*
- *nogle gange bliver det lidt uoverskueligt.  
Nogle gange mangler der måske et eksempel, som kan gøre tingene mere klare.*
- *Nogle gange var eksemplerne i helt så brugbare*
- *Bogen er for så vidt fint, men det kræver virkelig en veltilrettelagt undervisning, hvis man kun skal bruge bogen som primær kilde. Det synes jeg ikke har været tilfældet her.*
- *Eksemplerne på forskellige funktioner er for simple og det ville være dejligt med et ekstra lidt mere komplekst eksempel eller en bedre beskrivelse af funktionerne og hvordan de kan bruges.*

*Derudover mangler der også nogle funktioner som har været nyttige i afleveringerne, som f.eks. list.countby*

- *Der var ikke altid stor overensstemmelse mellem læsning og opgave*
- *Det eneste nyttige ved bogen har været at få viden om, hvilke funktioner, der eksisterer. Der er mange forklaringer og lange eksempler, som man godt kunne være foruden og kun giver støj. Samtidig er den meget mangelfuld ift. hvilke funktioner, der eksisterer, som vi har skullet bruge i vores afleveringer, så alt i alt har den ofte givet mere frustration end hjælp.*
- *Hvis man leder efter et svar til et specifikt problem*

*går man tit forgæves ved at lede i bogen.*

- *Den er lidt rodet og ustruktureret. Emnerne kom i en mærkelig rækkefølge. Emnerne bliver ikke forklaret godt nok til, at man kan forstå dem.*
- *Jeg synes der er ringe sammenhæng mellem øvelsesopgaver og bogensindhold.*
- *En del små fejl, dårlige formuleringer og nogle ting bliver ikke forklaret så godt. Jeg vil forslå, at have nogle studerende der lærer F# til at løbende skrive ned hvor de synes der er brug for forbedringer og mere dybdegående forklaringer.*
- *Bogen holder det ofte simpelt og overskueligt. Visse dele kan dog godt være forvirrende eller uoverskuelige for personer uden programmeringserfaring.*
- *Bogen virker for mig, som et leksikon man kan slå op i, hvis der er noget man er i tvivl om. Ikke særlig intuitiv at læse, som f.eks. bogen i DMA. Måske lidt for mange eksempler på kode.*
- *Det er noter. Der virker rodet. Der er ikke klar forklaring af syntax for basale ting.*
- *Tingene er forklaret med et dårligt sprog, og Jon begynder nogle steder at referere til og bruge ting, som først får en egentlig introduktion i kapitlet efter. Så det eksempel der skal forklares bliver totalt volapyk. Generelt føles det som om at der er en bestemt rækkefølge som tingene skal introduceres i for at koncepterne i bogen giver mening, men Jon glemmer denne rækkefølge en gang imellem.*
- *den er meget svær at finde rundt*
- *Bogen starter nemt og forståeligt ud, for en nybegynder til programmering. Men sværhedsgraden stiger alt for hurtigt midt i bogen.*
- *Lack of coherence between (some) concepts presented.*
- *Niveauet stiger meget hurtigt og kodeeksemplerne kan være svære at forstå.*
- *Det skrevne sprog er ikke grammatisk korrekt 1/10 gange, hvilket gør det svært at forstå nogle sammenhæng*
- *Eksemplerne har ikke en ordentlig sammenhæng mellem øvelsesopgaverne og afleveringsopgaverne. Kunne være godt med nogle eksempler der kan bruges i praksis*
- *For rodet  
For korte variabelnavne, hvilket gør det svært at forstå.*
- *brugte bogen, hvis jeg løb ind i problemer.*
- *Nogen gange kunne det være rart med flere eksempler af den samme ting i forskellige kontekst, da det kan hjælpe en med at se hvad der altid er der og hvad ændres, give en hjælp til forståelsen.*
- *Den er for tæt skrevet til en nybegynder. Jeg har ofte savnet flere eksempler med lidt forskellige vinkler. Et eksempel, og meget få på nettet, gør at man sidder har et meget lille*

*fundament for at løse opgaven.*

- *Mange gange kan det godt virke meget overvældende det der bliver skrevet. Og meget især i starten virker lidt som det bliver skrevet for tidligt (der står endda også at det først bliver forklaret 10 kapitler længere henne) og bidrager med til forvirring end til gavn. Kan være meget svært at huske alt der bliver skrevet, og nogle gange opstår "wall of text" problemet også.*
- *Det virker som en god og grundig gennemgang af F# for folk med en hvis programmeringserfaring, men virker rimelig overvældende hvis man starter ved nul. Selv om oversigten bliver gjort nemmere vha. stikord i margin, er der stadig instancer hvor lidt bladren bliver nødvendigt for at følge krydsreferencer til andre kapitler. Tingene bliver sædvanligvis kun forklaret én gang, hvilket for en nybegynder gør det mere kompliceret og tidskrævende at slå efter i tidligere kapitler. Især i starten af bogen (eller forelæsningerne) ville det have været en stor hjælp ikke bare at se funktioner som eksempler for sig selv, men som del af en større programblok, dvs. indbygget i andre funktioner - det er stadig svært for mig personligt at oversætte den ene kontekst til den anden.*
- *Eksemplerne var for simple.*
- *Skriv bogen, så den forklarer flere grundlæggende koncepter. Uheldigvis bliver der oftest slynget rundt med programmørjargon (lige fra start), som man kun forstår, hvis man har tidligere erfaring med programmering. Det er altså en bog, der er skrevet til programmører, hvilket er en uheldig tilgang. Datalogi bør være tilgængeligt for alle og ikke hovedsageligt folk, som man har en forventning om har skrevet kode før.*

*Skriv bogen på dansk. Vi er danskere, som tager en dansk uddannelse på et dansk universitet i Danmark. Modersmålet er nemmere at forstå for de studerende. Det er rart at kunne formidle sig på sit eget sprog i stedet for at være tvunget til at speake danglish in one's hverdag.*

- *Eksemplerne er virkelig ikke fyldestgørende i forhold til hvad vi skal bruge til at kunne lave de stillede opgaver.*
- *Lærebogen er, ærligt talt, ikke særligt velegnet som lærebog. Det er vigtigt at holde sig for øje hvad formålet med faget er. Er det at lære folk der allerede kan programmere at bruge F#, eller er det at lære folk at programmere. Det er uendeligt meget lettere at tilegne sig et nyt programmeringssprog når man allerede kan programmere end det er at lære at programmere til at begynde med. De fleste bilister kan vende sig til en ny bil på 5-10 min., men det tager meget længere at lære at køre bil i første omgang (og du skal have en kørelærer siddende ved siden af dig hele tiden). Sidstnævnte har brug for grundige og gennemskuelige forklaringer af stoffet. Bogen er, i sin nuværende form, mere refererende end forklarende, så der er behov for en langt grundigere gennemgang af sprogets syntaks, med fokus på forklaring af de enkelte nøgleords betydning. Førstnævnte, derimod, har mest af alt brug for et opslagsværk hvor de hurtigt kan finde syntaksen til det de skal bruge. Jeg har talt med flere der slet ikke læser bogen – kun bruger den som pensumliste – fordi de slår syntaksen op på nettet i stedet for. De vil altså heller ikke blive trukket ned af en bog der mere henvender sig til nybegyndere. Man skal huske at, det at opstille gode og illustrative eksempler, ikke er let. Et eksempel skal være let at følge (dette kræver også at man forklarer det trin for trin, i stedet for bare at opstille det) og skal illustrere hvad der foregår (jo flere mellemregninger, jo bedre); og ikke mindst skal de illustrere forskellen på forskellige metoder.  $2+2$  er det klassiske eksempel for addition, men  $2+2$ ,  $2*2$  og  $22$  giver alle det samme facit og illustrerer dermed ikke forskellen mellem de tre metoder. Man ville derfor let kunne forledes til at tro at de var ækvivalente. Af*

samme grund er det også vigtigt at have mere end ét eksempel for hver ting man gerne vil forklare. Dette hjælper også med at demonstrere den generelle anvendelighed af en metode, i stedet for kun at kunne anvende den på et smalt segment af problemer der ligner det stillede eksempel (man har måske nok lært at  $2+2=4$ , men man har ikke lært metoden + før man også kan håndtere  $7+4$ ). Jeg vil gerne fremhæve KBR-bogen fra Diskret Matematik og Algoritmer som et eksempel på en god lærebog, med masser af detaljerede og illustrative eksempler.

Det er meget typisk at man viser nogle meget simple eksempler og derefter stiller nogle langt mere komplicerede opgaver. Dette kan gøre det svært for den studerende at hente tilstrækkelig hjælp i eksemplerne. Det kan sammenlignes med at man fortæller hvordan man kan finde kvadratrødder ved systematisk at kvadrere heltal og se om det giver udgangspunktet, og demonstrere med kvadratrods 25; for herefter at bede de studerende finde kvadratroden til 117,2. Eller at demonstrere Pythagoras læresætning med sidelængderne 3 og 4; og herefter stille en opgave med sidelængderne 1 og 1. Det ville være bedre at gøre det omvendt! Opstil en række eksempler af stigende kompleksitet, hvoraf nogle overstiger kompleksiteten af de opgaver der vil blive stillet i starten. Når de studerende sidenhen er blevet fortrolige med metoden, kan man stille sværhedsgraden op over eksemplerne.

Jeg har nogle gange hørt snak om at man skal "hæve abstraktionsniveauet" som undskyldning for mere eller mindre uforståeligt undervisningsmateriale. Dette er totalt misforstået! Man skal derimod sænke abstraktionsniveauet – konkretisere så meget som muligt – ned til den studerendes niveau. Efterhånden som den studerendes forståelse forbedres, kan man øge abstraktionsniveauet i samme takt. "Abstraktionsniveau" henviser til en persons evne til at forstå komplekst stof. Alt hvad der overstiger dette niveau bliver simpelthen ikke forstået. Hvis du skal lære folk at tale kinesisk, nytter det ikke noget at du stiller dig op og taler kinesisk til dem. De eneste der får noget ud af det er dem der allerede kan kinesisk.

Nogle af bogens kapitler står også i en underlig rækkefølge. Fx forklares princippet i objektorienteret programmering først efter at objekter og klasser er blevet gennemgået. Dette burde nok være omvendt. Start oppefra og grad dig ned i detaljerne, i stedet for at forsøge at samle en masse spredte detaljer sammen til en helhed.

#### 1.8 Hvis du skal nævne POSITIVE aspekter af bogen, hvad skulle de så være?

- Der er meget lidt information om F# på nettet så bogen er uundværlig. Bogens mange eksempler gør tingene mere håndgribelige.
- Den er klart en af de bedste lærebøger, jeg er endnu stødt — uanset emne! Den virker kort og præcis, mens mere avancerede emne uden for bogens sigte trods alt blive nævnt, så et større perspektiv er til stede. Meget vel afbalanceret!
- Bogen går igennem de forskellige emner og man får et hurtigt overblik over hvad F# indeholder.
- gode eksempler, nemt at slå op i den
- Den kommer godt rundt om stoffet.
- Den går meget præcist igennem det den skal, hvilket er godt for så føler jeg ikke at man sidder og læser flere sider af ingenting. Hver sætningen er vigtig at læse.
- Den indeholdte det meste af den nødvendige viden.
- De fleste eksempler er ret gode

- *Der er massere af eksempler.*
- *Der er mange gode eksempler.*
- *ingen kommentare*
- *Den er pædagogisk og nem at følge, når man læser den.*
- *Det er godt at der findes eksempler specifikke ting.*
- *God opslags bog*
- *Gode eksempler og forklaringer. Oversigt over de forskellige moduler er nok det mest nyttige (Self om der mangler en del brugbare moduler)*
- *Jeg kan ikke komme i tanke om noget, for at være helt ærlig. Har ingen positive ting.*
- *Der er ingen.*
- *At den fås i PDF, og at den var gratis.*
- *Nogle af forklaringerne er helt exceptionelle, og kategoriseringen af emner er vel tænkt.*
- *Mange gode forklaringer, som man kan bruge i forbindelse med afleveringerne*
- *Eksemplerne er ret nyttige til tider.*
- *balancen imellem eksempler og tekst*
- *Der er mange eksempler, som gør det nemmere at skimme. Listen med funktioner til List (List.fold, List.reduce etc) var brugbar.*
- *Jeg oplevede at bogen udgjorde en ret solid base at lære f# ud fra. På trods af min tidligere kommentar, var jeg rigtig glad for afsnittet om lister, strings og array. Den var nem at slå op i, forklaringerne gjorde det nemt at finde den rigtige funktion til ens problem, og netop derfor ærgede det mig, at listen med indbyggede funktioner ikke var mere udvidet.*
- *letlæseligt sprog.*
- *Den er skrevet i LaTeX*
- *Relativt godt formidlet, godt med keywords i marginen og så var den selvfølgelig gratis :)*
- *Eksempler på kode og hvad de returnerer*
- *Den er kort og koncis i formuleringer. Størstedelen af tiden en god "kronologi" fra simpelt og nemt til avanceret og svært.*
- *Tabellerne var super fine og nyttige.*

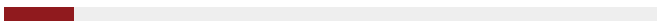
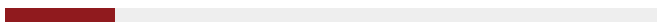
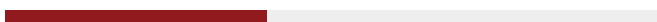
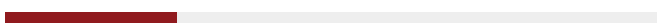
*God forklaring ved list modulerne*

- *God opslagsværk, gode forklaringer af moduler*
- *Opslagsværk og nogle enkle funktioner som er bedre forklaret end på internettet.*
- *Når der er korte og præcise eksempler som med nogle af list-funktionerne.*
- *Gode eksempler, let at forstå.*
- *Nogen af de indbyggede funktioner stod der, men vi ville godt have haft alle. Den var gratis.*
- *Flere eksempler på brugen af værktøjer.*
- *God struktur og flot opsætning. Nem at slå op i :)*
- *Den var gratis*
- *Fine eksempler*
- *Gode eksempler, og bogen er struktureret godt og giver mening ca. indtil kapitel 7*
- *den har gode eksempler*
- *Godt med mange og tydelige programmerinseksempler, og forklaringer på system.objects som List/string/array functions.*
- *The abundance of examples.*
- *Meget sammenfattende*
- *En masse om f# samlet i en bog. Vil være god at have hvis den var på tryk.*
- *Gode eksempler og sammenhæng mellem de forskellige emner.*
- *Basisinformationerne omkring alle de forskellige funktioner er super godt forklaret. Eksemplerne er også godt forklaret til de enkelte funktioner, hvis det blot er til forståelse.*
- *Tabeller*
- *Classes, list - array modullerne*
- *Den kommer gennem hele programmerings emnet godt, og jeg kan godt lide de tips de er highlighted så man ser dem, selvom man går lidt hurtigt over emnet. og det samme med stikordene i siden.*
- *Alt i den er forklaret præcist. Den har slet ikke været dårlig, men den har ikke, for mig, kunne stå alene.*



- *Generelt meget gode eksempler. Det er primært dem der har reddet forståelsen af mange af eksemplerne.*
- *Den virkede rimelig grundig og lister og tabeller af kommandoer og lign. var meget nyttige, selvom det på sigt var nemmere at bogmærke dem på internettet. Disse kunne evt. sammenfattes til hurtigere opslag som appendix.*
- *Formatteringen og de hyppige kodeeksempler gør den mere tilgængelig. Ofte skanner jeg efter kodeeksempler.*
- *Flot font ;)*
- *Bogen virker som et ærligt forsøg på at skrive en lærebog, den er bare ikke særligt vellykket. Med en gennemgribende omskrivning kunne den dog godt blive et værdifuldt indlæringsværktøj.*

1.9 Hvor stor nytte har bogen været for din læring på kurset?

Meget lille	10	10,5 %	
Lille	16	16,8 %	
Acceptabel	38	40,0 %	
Stor	25	26,3 %	
Meget stor	6	6,3 %	