

Techn. Fakultät - Erwin-Rommel-Str. 60 - 91058 Erlangen

Prof. Dr. Michael Philippsen
(PERSÖNLICH)

SS10: Auswertung für Parallele und Funktionale Programmierung (10s-PFP)

Sehr geehrter Herr Prof. Dr. Philippsen,

Sie erhalten hier die Ergebnisse der automatisierten Auswertung der Lehrveranstaltungsevaluation im SS 2010 zu Ihrer Umfrage:

- Parallele und Funktionale Programmierung (10s-PFP) -

Es wurde hierbei der Fragebogen - v_s10 - verwendet, es wurden 37 Fragebögen von Studierenden ausgefüllt.

Die Note 1 kennzeichnet hierbei eine maximale Güte, die Note 6 eine minimale Güte für die einzelnen Fragen bzw. Mittelwerte.

Auf der nächsten Seite zeigt der zuerst angegebene "Globalindikator" Ihre persönliche Durchschnittsnote über alle Indikatoren/Kapitel, deren Noten danach folgen.

Für die Ergebnisse aller Einzelfragen werden je nach Fragen-Typ die Anzahl und Verteilung der Antworten, Mittelwert und Standardabweichung aufgelistet.
Die Text-Antworten für alle offenen Fragen sind jeweils zusammengefasst.

Auf der letzten Seite befindet sich eine Profillinie im Vergleich zu den Mittelwerten aller Rückläufer für diesen Fragebogen-Typ. Die Profillinie eignet sich auch zur Präsentation in der LV.

Eine Einordnung Ihrer Bewertung ist nach Abschluss der Ergebnisauswertung unter <http://eva.uni-erlangen.de> (--> Technische Fakultät --> Ergebnisse --> SS 2010) möglich, hierzu die Bestenlisten, Percentile, etc. einsehen.

Bitte melden Sie an eva@techfak.uni-erlangen.de die Anzahl der ausgegebenen TANn, wenn Sie das bis jetzt versäumt haben.

Mit freundlichen Grüßen

Bernhard Schmauß (Studiendekan, schmauss@lhft.eei.uni-erlangen.de)
Jürgen Frickel (Evaluationskoordinator, eva@techfak.uni-erlangen.de)



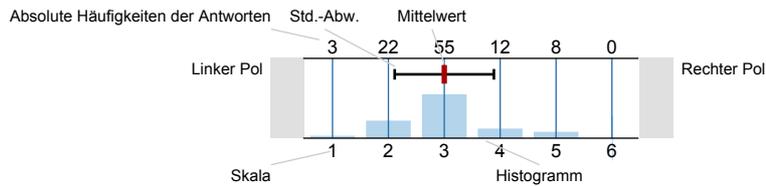
Prof. Dr. Michael Philippsen
 Parallele und Funktionale Programmierung (10s-PFP) (10s-PFP)
 Erfasste Fragebögen = 37

Globalwerte

Globalindikator		mw=1.83 s=0.93
Vorlesung im Allgemeinen		mw=1.9 s=0.9
Didaktische Aufbereitung		mw=2.19 s=1.18
Persönliches Auftreten des Dozenten		mw=1.54 s=0.79
Verwendete Hilfsmittel		mw=1.89 s=1.11
Gesamteindruck		mw=1.64 s=0.68

Legende

Frage text



n=Anzahl
 mw=Mittelwert
 s=Std.-Abw.
 E.=Enthaltung

Klick on british flag to get the english survey
 Achtung: Beim Anklicken der Sprachsymbole verlieren Sie alle bisherigen Eintragungen !!

Allgemeines zur Person

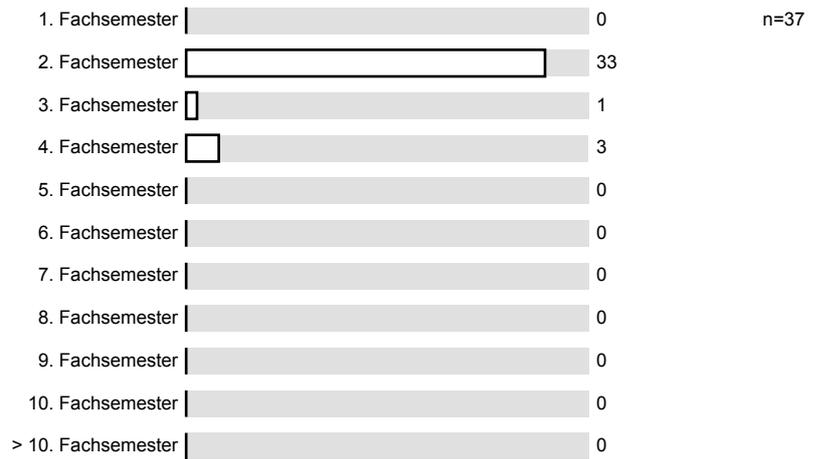
^{2_A)} Ich studiere folgenden Studiengang:

INF - Informatik	<input type="text" value="25"/>	25	n=37
INFL - Informatik für Lehramt	<input type="text" value="3"/>	3	
(T)Math - (Techno)-Mathematik	<input type="text" value="3"/>	3	
WINF - Wirtschaftsinformatik	<input type="text" value="6"/>	6	

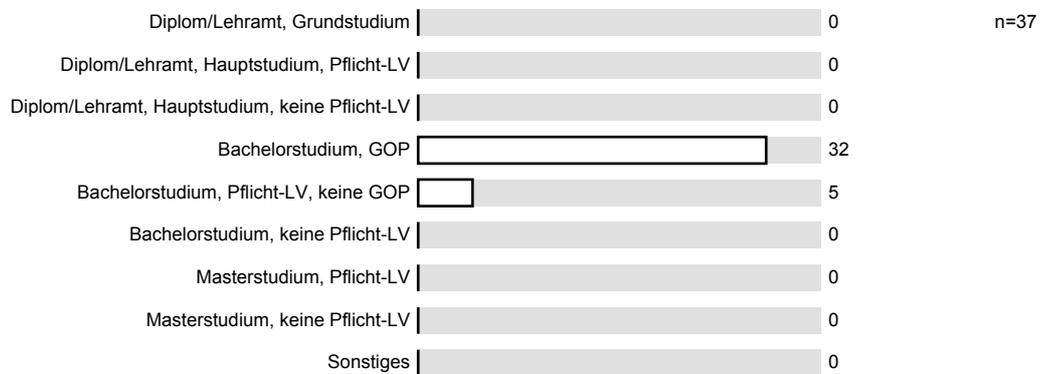
^{2_B)} Ich mache folgenden Abschluss:

Dipl. - Diplom	<input type="text" value="0"/>	0	n=37
B.Sc. - Bachelor of Science	<input type="text" value="34"/>	34	
M.Sc. - Master of Science	<input type="text" value="0"/>	0	
Staatsexamen	<input type="text" value="3"/>	3	
Dr.-Ing. - Promotion	<input type="text" value="0"/>	0	
PhD - Doctor of Philosophy	<input type="text" value="0"/>	0	
Dipl.-Ing. mit Zusatzzertifikat	<input type="text" value="0"/>	0	
M.Sc.(hons) - Master of Science with Honours	<input type="text" value="0"/>	0	
Zwei-Fach-Bachelor of Arts	<input type="text" value="0"/>	0	
Sonstiges	<input type="text" value="0"/>	0	

2_C) Ich bin im folgenden Fachsemester:

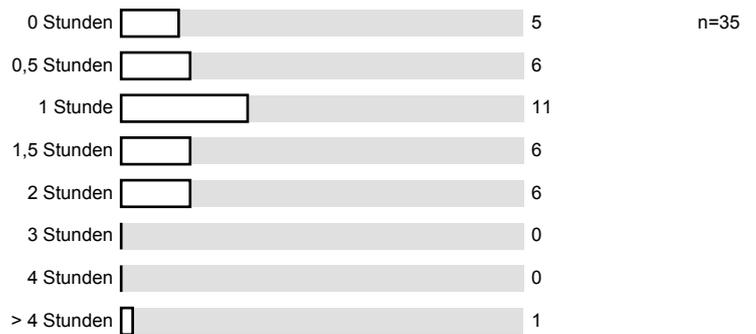


2_D) Diese Lehrveranstaltung gehört für mich zum



Mein eigener Aufwand

3_A) Mein Durchschnittsaufwand für Vor- und Nachbereitung dieser Vorlesung beträgt pro Doppelstunde (90 Min.):

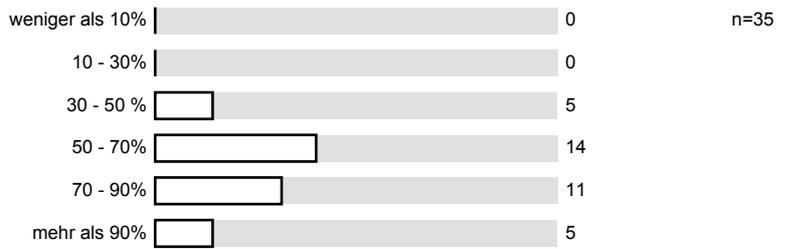


3_B) Ich besuche etwa . . . Prozent dieser Vorlesung.

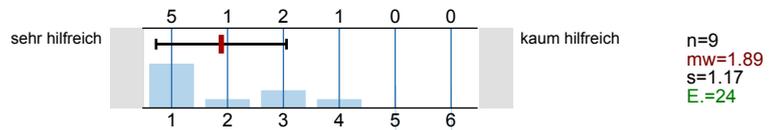


Durchführung

4_A) Der oben aufgeführte Dozent hat diese Vorlesung zu . . . selbst gehalten.

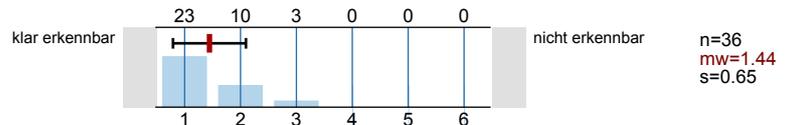


4_B) Die evtl. zusätzlich angebotenen Tutorien (nicht die regulären Übungen!) sind

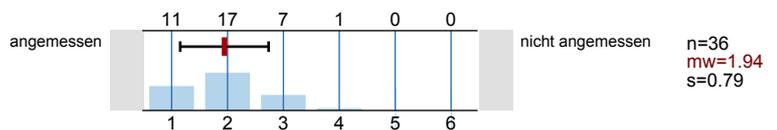


Vorlesung im Allgemeinen

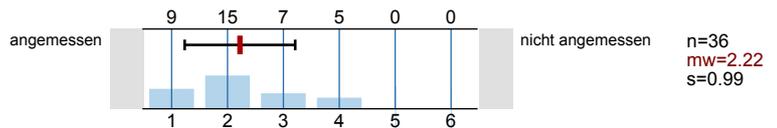
5_A) Zielsetzungen, Struktur und Schwerpunkte des Vorlesungsinhalts sind:



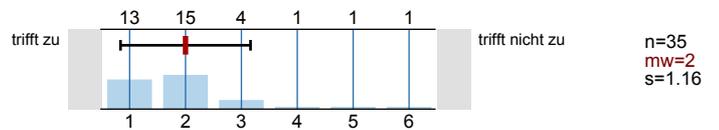
5_B) Der Umfang des Stoffes ist:



5_C) Der Schwierigkeitsgrad des Stoffes ist:

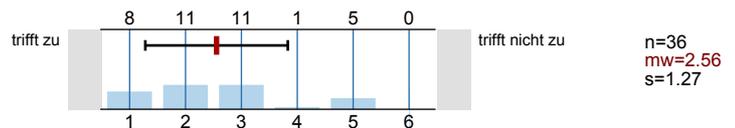


5_D) Zusammenhänge und Querverbindungen zu anderen Studieninhalten werden deutlich aufgezeigt.

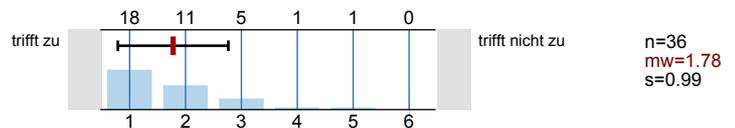


Didaktische Aufbereitung

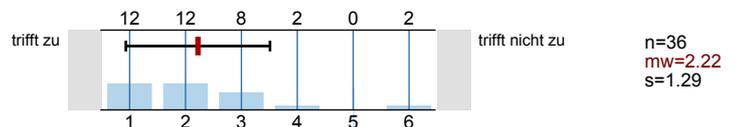
6_A) Der dargebotene Stoff ist nachvollziehbar, es ist genügend Zeit zum Mitdenken vorhanden.



6_B) Der rote Faden ist stets erkennbar.

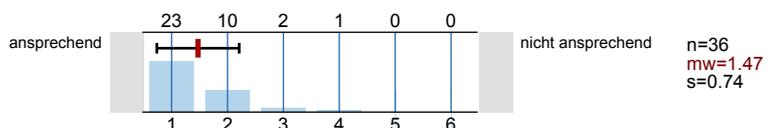


6_C) Der Bezug zu Übungen und Prüfungsanforderungen wird hergestellt.

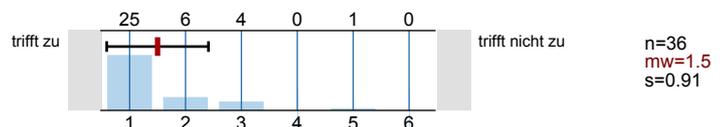


Persönliches Auftreten des Dozenten

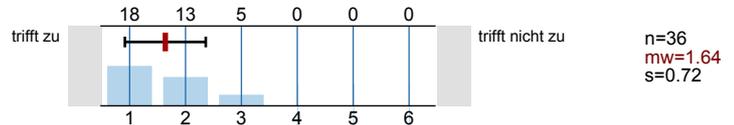
7_A) Der Präsentationsstil des Dozenten ist:



7_B) Der Dozent weckt das Interesse am Stoff.

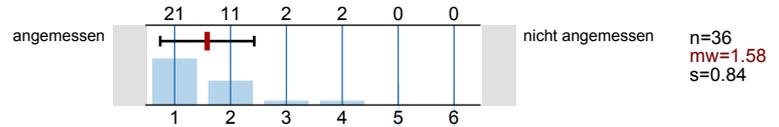


7_C) Der Dozent vergewissert sich, dass der Stoff verstanden wurde und geht gut auf Zwischenfragen ein.

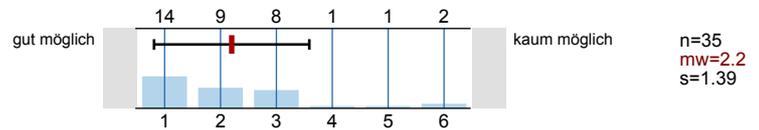


Verwendete Hilfsmittel

8_A) Der Einsatz von Medien (Tafel, Overhead-Projektor, Beamer, etc.) ist:

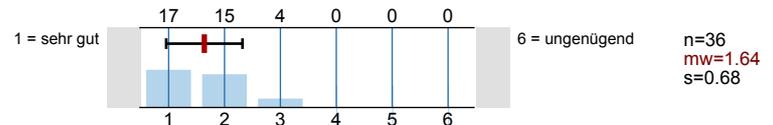


8_B) An Hand des zur Verfügung gestellten Begleitmaterials und der Literaturhinweise sind Vor- und Nachbereitung:



Gesamteindruck

9_A) Insgesamt bewerte ich die Vorlesung mit der Note:



Weitere Kommentare

10_A) An der Lehrveranstaltung gefällt mir besonders:

- , dass der Dozent den Stoff ansprechend herüberbringt. Außerdem wird versucht eine klare Linie (sprich roter Faden bzw. Zusammenhänge) zu fahren. Die Theorieaufgaben der Übungsblätter sind sehr gut nachzuvollziehen und bereiten gut auf Vorlesung bzw. Klausur vor.
- - Die Sprechgeschwindigkeit von Prof. Philippsen, immer noch so amüsant wie in AuD :)
 - Die guten Beispiele, die zusätzlich zu den in den Folien abgedruckten an der Tafel skizziert und erklärt werden.
 - Die Tatsache, dass hier eine sehr aktuelle Thematik und eine sehr wichtige Grundlage für "modernes Programmieren" schon im 2. Semester angesprochen wird, ist sehr sehr lobenswert.
- - aktuelle Relevanz: Klasse, kann man (spätestens) nach der Uni gut gebrauchen
 - trotz schneller Sprechweise gut verständlich
 - hohes Know-How, gute Aufbereitung der Folien
 - Vorbildhafter Dialog mittels EST.
- - man merkt das Prof. Dr. Phillippsen ahnung vom stoff hat
 - vorlesungsstil von Prof. Dr. Phillippsen
- @Philippsen: Der mitreisende Präsentationsstil (Schade dass nicht alle Veranstaltungen von Ihm gehalten wurden)
@Oster: Passt schon
- Alles top, macht spaß zuzuhören
- Beide Profs konnten gut erklären.
- Der "funktionale Programmierung" Teil und die Grundlagen des parallelen Programmierens.
- Die "hellblaue" Farbe hat sich deutlich verbessert :)
Fehler (auf den Folien) sind schnell korrigiert
Im Allgemeinen die Vorlesung gefällt mir ganz gut, weiter so!
- Die Folien, der Präsentationsstil, die kompetenten und netten Dozenten, die auch jederzeit begeistert auf Zwischenfragen eingehen.
- Die Parallelen Probleme werden gut erörtert und anschaulich geloeszt.
- Die Vorlesungen von Prof.Philippsen sind die lustigsten. Auch wird der Stoff von ihm sehr gut und ansprechend erklärt.
- Ein guter Überblick über das Thema wird verschafft, guter Präsentationsstil
- Erklärung von Prof. Phillippsen sind sehr ansprechend und verständlich.
Die Sprechgeschwindigkeit passt - zumindest für den AuD-hörer - genau!
Auch andere Dozenten haben die Vorlesung ganz gut gehalten!
- Gut strukturiert, schwierige Themen sind meist gut erklärt

- In der Ersten VL haben Sie uns wirklich gut und interessant erklärt wieso wir Parallele Programmierung brauchen. Ihre Folien sind nach wie vor die am besten strukturierten die ich kenne, was natürlich auch für die Qualität gilt.
- Lebendiger Präsentationsstil und übersichtliches, aber trotzdem vollständiges Folienlayout lassen nichts zu wünschen übrig (gilt sowohl für Java als auch für Haskell). Die Beispiele aus der Realität zur Parallelisierung tragen ihren Teil zum Verständnis bei, vor allem die Fehler.
- Motivierte & Motivierende Dozenten!
- PFP ist die spannendste Vorlesung meines 2. Semesters. Endlich geht's in die Tiefe. Toll, dass wir was über Multicore lernen!
- Philippsen: der höchst informative und unterhaltsame Vortragsstil. Man hat sehr viel verstanden, was vielleicht auf den ersten Blick recht kompliziert schien und hat auch viel Hintergrundinformation erhalten.
Oster: Endlich kann man in der Informatik die Dinge mathematisch aufschreiben:-)))
- Prima - als Winf-Student hatte ich erst Angst vor der Vorl. Ist aber superspannend - ich komme sogar mit. Mal sehen wie lange ich's beim Funktionalen Teil noch blicke
- Schon allein der Präsentationsstil der Dozenten macht die Vorlesung zur besten Vorlesung im 2. Fachsemester. Auch solch trockene Themen wie das Lambda-Kalkül werden interessant und kurzweilig dargestellt.
- Sehr interessantes Themengebiet und meiner Meinung nach auch sehr gut dargestellt.
- die folien sind sehr gut um den stoff zu hause nochmal durch zu arbeiten
- erwartungsgemaess tiptop! Noch besser als AUD. Zumindest bis heute. Bin auf den F-Teil gespannt.
- gewohnte Philippsen-Qualität ;) Oster auch sehr guter Vortrag beide angemessener Humor
- gutes Skript
- schöne Anekdoten, angenehmer Still
- von vorne bis hinten interessant und didaktisch sehr gut
- witziger Vortrag

^{10.B)} An der Lehrveranstaltung gefällt mir Folgendes weniger, und ich schlage zur Verbesserung vor:

- - auf noch öfter 'mal die gute alte Tafel benutzen...
- Vorlesungsinhalte von Haskell wirken wie 'rein gequetscht. Vielleicht schon eher in die Vorlesung mit einbauen und Niveau langsamer steigern
- - haskell wirkt sinnlos(welcher kunde bezahlt für die funktion eines programms, das in haskell programmiert ist? welche sinnvollen sachen kann man mit haskell machen?) -> (weglassen?)
- das meckern von Prof. Dr. Phillippsen ist manchmal zum fremdschämen, ich finde es nicht notwendig worte darüber zu verlieren wenn jemand zu spät kommt, das hält nur auf und unterbricht die gedankengänge!
- petrinetze sind auch irgendwie sinnlos. davon mal gehört zu haben ist ja ok, aber dafür könnte man ruhig wehniger zeit opfern und anstelle lieber die schweren themen um die dadurch gesparte zeit verlängern.
- Auf der Anwendersicht überflüssig, für die Systemsicht weit zu oberflächlich -- warum sowas im 2. Semester?
LISP ist *nicht* eine (rein) funktionale Sprache! Mag für Scheme stimmen, Common LISP ist genauso Objektorientiert und Imperativ
- Da keine Übungs-Tan erhalten: bitte dringend Tutoren überprüfen.
- Den Teil "Anwendung der parallelen Programmierung" fand ich weniger spannend. Ich hätte es mir gewünscht etwas weniger Java Beispiele zu sehen, und dafür den Haskell Teil ein wenig zu erweitern.
- Der Tutor [REDACTED] war leider eine Pfeife. Hatte sich nie vorbereitet, hatte keine Ahnung vom Stoff, verstand sogar die Musterlösung teilweise nicht. Die Übung beschränkte sich meistens auf warten, bis der Tutor zu Ende gedacht hatte und abschreiben der Musterlösung, die Programmieraufgaben wurden nur vorgelesen ohne irgendwelche Lösungshinweise zu geben. So wird Uni zur Qual!
- Die vielen Grafiken sind für die Vorlesung super aber für die Nachbereitung ein wenig informationslos. Vielleicht könnte man noch ein paar Zwischenfolien einfügen, die die Erklärung beinhalten und in der Vorlesung einfach übersprungen werden.
- Ein Tutorium für die Klausur wäre nicht verkehrt.
- Herr Dotzler hat bei der ersten Vorlesung leider nur die Folien vorgelesen, weshalb die Vorlesung relativ nutzlos war. Dies hat sich aber in den darauf folgenden Vorlesungen geändert. Allgemein hätte man klarer herausstellen können, wann Variablen in den Registern des Prozessors liegen und wann im Arbeitsspeicher. Gerade habe ich den Eindruck, dass Haskell am Ende des Semesters "reingedrückt" wird, keine Zeit besteht den nicht so einfachen Stoff langsamer durchzugehen. Herr Oster gibt sich Mühe, aber für Jemanden der

darüber das erste mal etwas hört ist die Sache schwierig und es braucht etwas mehr Zeit sich Klarheit zu verschaffen.

- Ich wuerde mir einen groeszeren Funktionalen Anteil wuenschen, und dass dieser nicht so stiefmuetterlich behandelt werden wuerde. (Nur weil das Semester kuerzer ist muss man ja kein Haskell weglassen, es wuerde reichen ein oder zwei parallele Sachen wegzulassen).
- Leider kein Übungs-Tan vorhanden, deshalb hier Kritik an Übung [REDACTED] hat Stoff selbst nicht ganz verstanden, hangelt sich sehr langsam an der Musterlösung entlang, langes warten, pnriegmatisches zuruckhaltendes Auftreten. Vorbesprechung der sehr wichtigen Programmieraufgaben mangelhaft
Anfangs keine Korrektur der Übungsabgaben (scheinbar wg. EST-Fehler(?))
kein Zurücksetzen der Abgabemöglichkeit nach Bekanntwerden des Fehlers.
Es sollte überlegt werden, ob Musterlösungen der Übungsaufgaben herausgegeben werden.
AUD Korrektur und Vorbesprechung wesentlich zielgerichteter
- Mangel an Pflichtabgaben verführt zum "Bummeln", da trifft aber die Vorlesung keine Schuld
- Mehr Funktional und nicht soo viel Parallel.
- Mir gefällt besonders nicht, dass die Vorlesung in fast keiner Hinsicht auf die praktischen Aufgaben der Übungen vorbereitet und ich deshalb sehr viel Zeit in Nachbereitung stecken muss, um diese lösen zu können (wenn überhaupt möglich). Manche Aufgaben sind derart weit hergeholt, dass nicht mal durch Internetrecherchen ähnliche Sachverhalte aufzufinden sind, so dass ich teilweise völlig ratlos der Aufgabe gegenüberstehe. Folglich muss ich oft den Übungsleiter um Rat fragen, allerdings ist dies sehr Zeitaufwendig, ich habe auch noch andere Fächer, die ich Vor- und Nachbereiten muss, nicht nur PFP. Da PFP eine GOP ist, ist dies sehr stressig, da ich diese ja bis zum 3. Semester bestanden haben muss. Darüber hinaus vermitteln mir die praktischen Übungsaufgaben nicht das Gefühl den Stoff komplett und richtig verstanden zu haben. Im Internet findet man viele Seiten, die verstärkt auf konkrete Problemstellungen eingehen, inklusive Beispielcode, allerdings sind auch meist damit die Übungsaufgaben nicht lösbar. In den Übungen selbst wird auch kaum auf die praktischen Übungsaufgaben eingegangen, was ich sehr schade finde, soll ich mir das etwa dann in den Ferien selbst beibringen, nachdem ich die Klausur bestanden/nicht bestanden habe? Die praktischen Übungsaufgaben sollten strukturierter zum Ziel führen, die Anzahl an Übungsaufgaben ist zwar mehr als ausreichend, allerdings sind diese dann sehr zeitaufwendig. Evtl. wären kleine "Häppchen" an Übungsaufgaben sinnvoller, die sich dann konkret auf bestimmte Probleme spezialisieren, so dass man an einer Aufgabe nicht stundenlang sitzen muss, sondern diese innerhalb von 1-2 Stunden lösen kann. Als Zusatz könnte man dann eine oder zwei komplexere Aufgaben stellen, die dann zusätzlich zur Vertiefung und Festigen des Stoffes dienen sollen.
Zum Skript: Das Skript ist zwar schön strukturiert und mit vielen Bildern ausgestattet, allerdings helfen mir die Codesnippets so gut wie garnicht in den Übungsaufgaben, mit Google findet man da wesentlich bessere und anschaulichere Außerdem ist das Skript an vielen Stellen überladen, so dass ich teilweise garnicht mehr den Überblick behalte, um was es überhaupt gehen soll. Da schlage ich vor, den Sachverhalt bzw. das Problem anhand eines konkreten und offensichtlichen Beispiels Schritt für Schritt zu erklären, am besten gleich inklusive sinnvollen Codesnippets, so dass alles nachzuvollziehen ist. Konkret: Client & worker finde ich katastrophal, dank Wikipedia hab ich das dann verstanden. Gegenbeispiel: MapReduce finde ich im Skript Top erklärt, so sollte es sein (!), v.a. sieht man dank übersichtlicher Skizze, was da überhaupt passiert.
Zur Vorlesung: Ich finde es zwar toll, dass die Vorlesung meist (sehr) vorzeitig beendet ist, da wir im Stoff so schnell vorankommen, allerdings ziehen Sie den Stoff so schnell durch, dass wir nichtmal Zeit zum Nachdenken haben, ergo: es werden keine Fragen während der Vorlesung gestellt^^ Evtl. könnte man da produktive Einschübe, z.B. mit extra Beispielen, nachrüsten, um das Verständnis zu vertiefen oder auf Probleme hinweisen.
Zu den Übungen: Da muss was geschehn. Es kann nicht sein, dass die praktischen Übungsaufgaben so gut wie garnicht angesprochen werden, zumindest die konkreten Schwierigkeiten einer Aufgabe sollten nochmal (und wenns nur kurz ist) erklärt werden.
Noch was ganz anderes: Das der Termin für die PFP-Klausur so früh stattfindet, ist sehr ärgerlich, da wir am 27. Juli noch KonzMod Klausur schreiben, sprich die Wahrscheinlichkeit beide Klausuren (die ja beide GOP sind und ich deshalb NICHT aufs 3. Semester verschieben kann, da ich mir dadurch ja einen Versuch verschenken würde, aufs 4. Semester darf ja nichts verschoben werden) zu bestehen bzw. gut zu bestehen, geht gegen Null. Inwiefern Sie was dafür können, weiß ich nicht, ich wollte das jeder noch als Randnotiz erwähnen, evtl. kann man da ja (zukünftig wenigstens) was unternehmen.
- Oster sollte etwas dynamischer, extrovertierter werden. Er soll sich seinen Chef als Vorbild nehmen.
- Philippsen: Die Zeitaufteilung zwischen (trivialer) Theorie und der Umsetzung im Code war manchmal etwas ungünstig. So konnte man sich zwar mehrfach anhören, was gewisse Konstrukte wie Thread-Pools, Executor, etc. machen oder wie der Algorithmus von Dijkstra funktioniert (--> AuD;-)), aber wie man das dann konkret programmiert und auf was man da zu achten hat, das ging manchmal unter. Vielleicht liegt es aber auch einfach nur am Untalent des gemeinen Mathematikers, die Interface- und Methodenbibliotheken und -strukturen nicht sofort zu durchleuchten, bzw. nicht zu wissen worauf besonders zu achten ist;
- Wie üblich wäre anzumerken, dass es schön wäre wenn Herr Philippsen etwas Geschwindigkeit aus seiner Vorlesung nehmen würde. Es ist zwar nett, dass man bereits nach 55 Minuten fertig ist, aber es wäre viel netter wenn man nach 90 Minuten rausgeht und noch viel mehr verstanden hat als beim üblichen Folienmarathon.
- Wäre schön gewesen wenn sie öfters da gewesen wären Herr Philippsen.
Herr Oster, bitte in Ihren Folien auch wie Herr Philippsen auf Toner sparen achten (schwarze Felder).
Herr Oster, Lamda Kalkühl... das BSP mit True False und dann AND dort haben Sie in Gedanken eingesetzt, bitte einfach auf einer Folie schritt für schritt zeigen.
- es fällt mir sehr schwer aufzupassen, obwohl ich meist ausgeschlafen bin. es ist sehr anstrengen immer nur auf den beamer zu starren. vielleicht würde eine gelegentliche tafelschrift zum mitarbeiten/besserem aufpassen helfen.
- gibt aus meiner sicht nichts zu verbessern
- keine kompletten Beispiele zu callable und future Objekten, so dass man keine Chance hat diese zu benutzen (wurden außerdem in der Vorlesung viel zu kurz behandelt).
Folge: Man muss sich alles selber beibringen, obwohl man dafür eigentlich keine Zeit hat(und auch keine Studiengebühren bezahlen müsste!)
- zu langsames Sprechtempo

^{10,C)} Zur Lehrveranstaltung möchte ich im Übrigen anmerken:

- -
- Bitte nicht den Humor verlieren, ist oft entscheidender Faktor zum wach bleiben. Und als Kölner gefällt mir besonders immer wieder auf die '4711' zu treffen, selbst in Arrays als '4 | 7 | 11' (:
- Da der Haskell-Teil sehr komplex ist, wäre es eventuell gut, einen Teil des Stoffes wegzulassen, damit der Rest nicht ganz so "kompakt" abgehandelt werden muss.
- Das mit der "Hörsaalregel" wirkt oft sehr fehl am Platz. Im Hörsaal ist es fast immer leise und auf jeden Fall immer so leise, dass man alles gut verstehen kann.
- Die Lambda-Kalkül Vorlesung war die beste Vorlesung im Semester.
- Die Übungsaufgaben (habe die Übungen nicht besucht, sondern nur abgegeben) wuerden mehr spass machen, wenn man durch die parallelisierung einen echten benefit sehen wuerde und nicht nur um des Konzepts Willen durchgezogen werden wuerden. Achja und das Java MapReduce ist echt nicht schoen. Seltsame API
- Die Vorlesungen waren sehr amüsan, v.a. dank Prof. Philippsen's Witzen zwischendurch ^^
- Die schnelle Sprechweise muss wohl am Lehrstuhl liegen, auch wenn Herr Oster noch 2-3 Gänge hinter Herrn Philippsen zurückliegt. Zum Glück gewöhnt man sich schnell dran.
- Die sehr gut und nachvollziehbar gestalteten Folien. Weiter so!
- Die zweite Funktional-vorlesung ist längst nicht so schlimm wie Anfangs angedroht.
- Ich wollte mit "sehr gut" Evaluieren, und dann kamen die Schönfinkelisatonen... :)
- Studenten mit Fußfetisch kamen an heißen Sommertagen bestimmt auf ihre Kosten... :)
- Leider sind die Programmierübungen nicht zwingend - das verführt leider dazu, daß man sie doch nicht macht.
- Pflichtabgaben hätte mir persönlich nicht geschadet, obwohl ichs so gannt toll fand :)
- Philippsen for Unirektor!
- Sehr interessantes Konzept, und für die Zukunft der Informatik sicher sehr hilfreich. Kann man auf jeden Fall weiterempfehlen;-)
- Trotz allen Anmerkungen anderer Studenten der Dozent spräche zu schnell, möchte ich sagen, dass ich finde, dass die Sprechgeschwindigkeit so passt.
- Vortragsstil von Herrn Phillippsen motivierte zum Mitdenken, sodass man am Ende der Vorlesung den groben Ablauf derselben wiedergeben kann. Bei Herrn Dotzler waren die Vorträge leider weitestgehend monoton, sodass es mir während der Vorlesung sehr schwer fiel, dem Geschehen zu folgen, das heißt, dass ich meistens nach 15 Minuten nicht mehr wusste, worum es vor 15 Minuten ging. Teilweise ging er jedoch näher auf die Folien ein und zeigte nicht sofort offensichtliche Zusammenhänge auf, was mich glücklicherweise meistens wieder aus meiner Starre gerissen hat.
Herr Osters Vortragstil ist bisher in Ordnung, der Stoff wurde verständlich erklärt. Die Lambda-Kalkül Vorlesung sollte jedoch besser auf der Tafel erfolgen, fertige Rechnungen auf Folien vorzulesen ist aus meiner Sicht didaktisch eher ungünstig. (Wobei mir durchaus bewusst ist, dass auch das separates Einüben nicht ersetzt.)
- Wenn das Sommersemester 1 Woche weniger hat, warum kürzt man dann beim kleineren Teil?
- es ist VIEL ZU KALT im hörsaal!!!
draußen hat es 36° und ich ziehe mich dementsprechend an, wieso hat es dann im hörsaal nur noch (wenn überhaupt) 20-25°???????????????? das macht absolut keinen sinn!!
ich muss manchmal sogar früher gehen weil ich da drin sonst erfriere!!
- es wäre schön, wenn es wenigstens zu jedem wichtigem Thema ein ausführliches Beispielprogramm z.B. als download gäbe, da es ja keine Musterlösungen gibt; Alternative: MUSTERLÖSUNGEN!!!
(die Ausrede, dass dann nur noch die Musterlösungen für die Klausur auswendig gelernt werden würden, ist mir viel zu kindisch, da man sich dabei ja nur selbst betrügen würde!!!)
- im allgemeinen SEHR interessantes thema. war die beste vorlesung dieses semester(bis auf haskell).
- immer ein schöner und interessanter Abschluss der Woche
- top-notch ;)

Optionale Zusatzfragen des Dozenten

^{11,A)}

Es wird keine Auswertung angezeigt, da die Anzahl der Antworten zu gering ist.

11_B)	Es wird keine Auswertung angezeigt, da die Anzahl der Antworten zu gering ist.
11_C)	Es wird keine Auswertung angezeigt, da die Anzahl der Antworten zu gering ist.
11_D)	Es wird keine Auswertung angezeigt, da die Anzahl der Antworten zu gering ist.
11_E)	Es wird keine Auswertung angezeigt, da die Anzahl der Antworten zu gering ist.
11_F)	Es wird keine Auswertung angezeigt, da die Anzahl der Antworten zu gering ist.
11_G)	Es wird keine Auswertung angezeigt, da die Anzahl der Antworten zu gering ist.
11_H)	Es wird keine Auswertung angezeigt, da die Anzahl der Antworten zu gering ist.
11_I)	Es wird keine Auswertung angezeigt, da die Anzahl der Antworten zu gering ist.
11_J)	Es wird keine Auswertung angezeigt, da die Anzahl der Antworten zu gering ist.

Profillinie

Teilbereich: Technische Fakultät (TF)
 Name der/des Lehrenden: Prof. Dr. Michael Philippsen
 Titel der Lehrveranstaltung: Parallele und Funktionale Programmierung (10s-PFP)
 (Name der Umfrage)

Vergleichslinie: Mittelwert aller Vorlesungs-Fragebögen
 im SS10

