Lehrevaluation Wintersemester 2010/11

Informatik I

Name des Dozenten: PD Dr. Jan-Georg Smaus

Name der Tutoren:

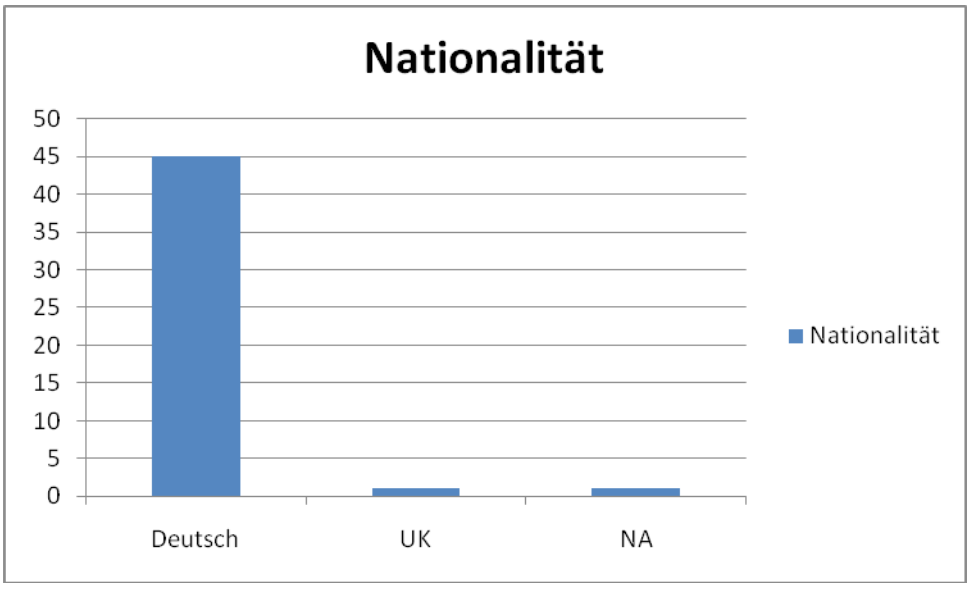
Unterrichtssprache in der Vorlesung: deutsch

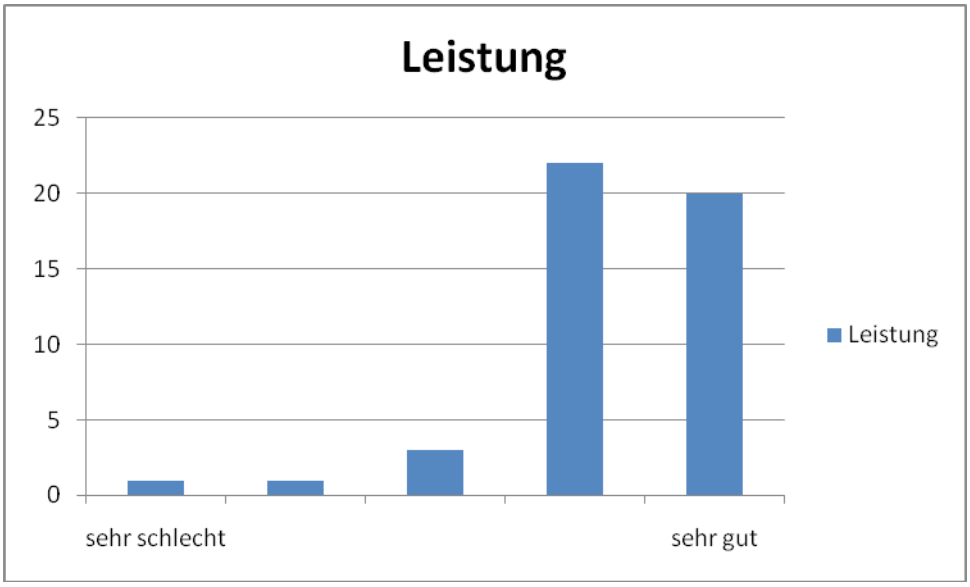
Ausgewertete Fragebögen: 47

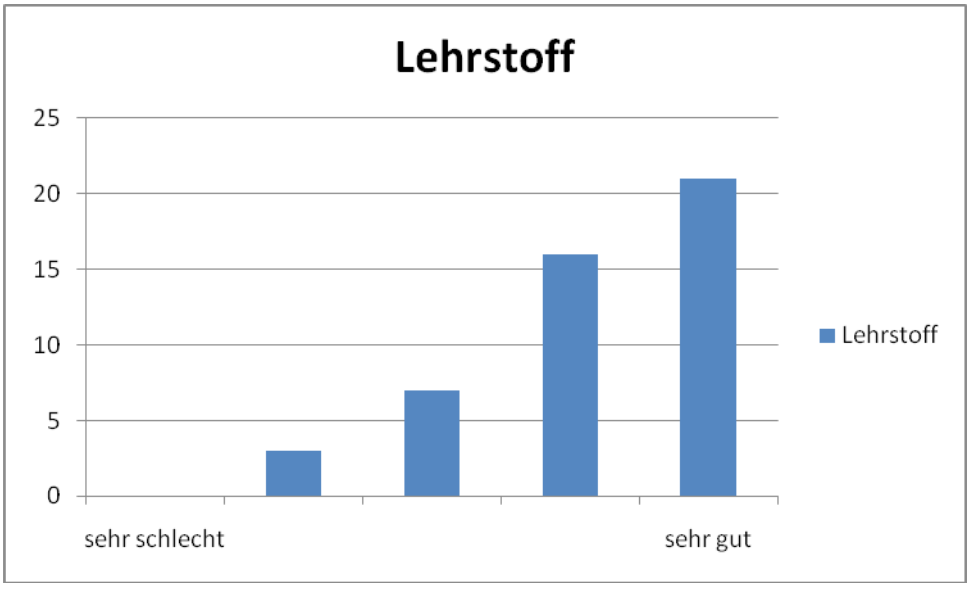
Lehrpreisvorschläge: 14

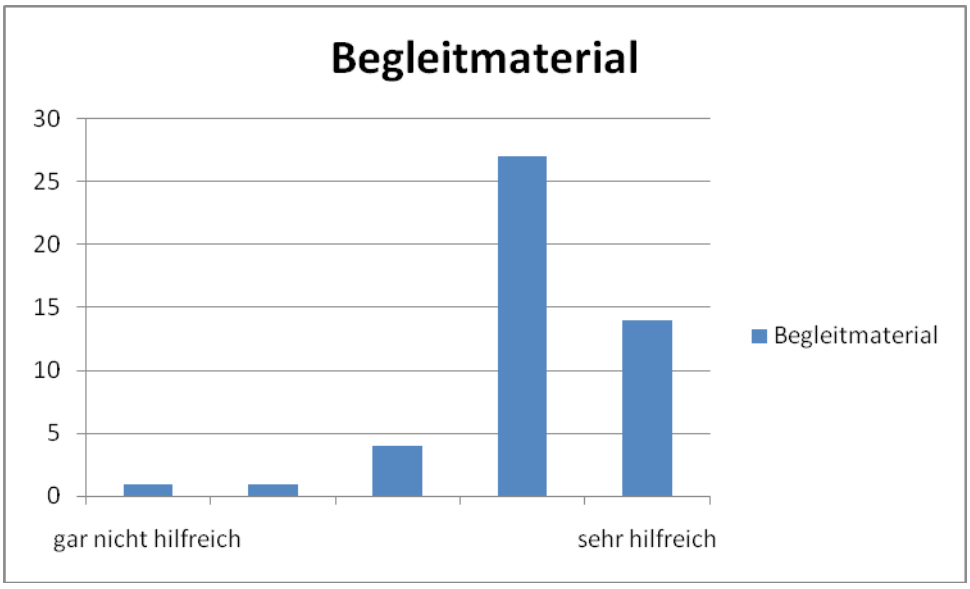
Aufbau der Auswertung:

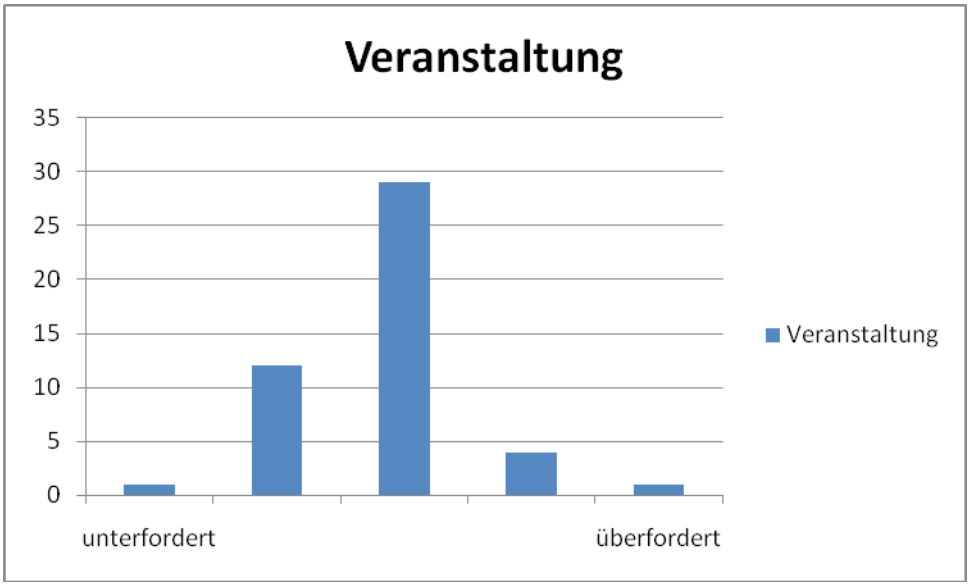
- Allgemeine Informationen zur Veranstaltung
- Diagramme einiger ausgewählter Fragen
- Tabellarischer Überblick über die statistischen Ergebnisse
- Auflistung der Freitext-Kommentare

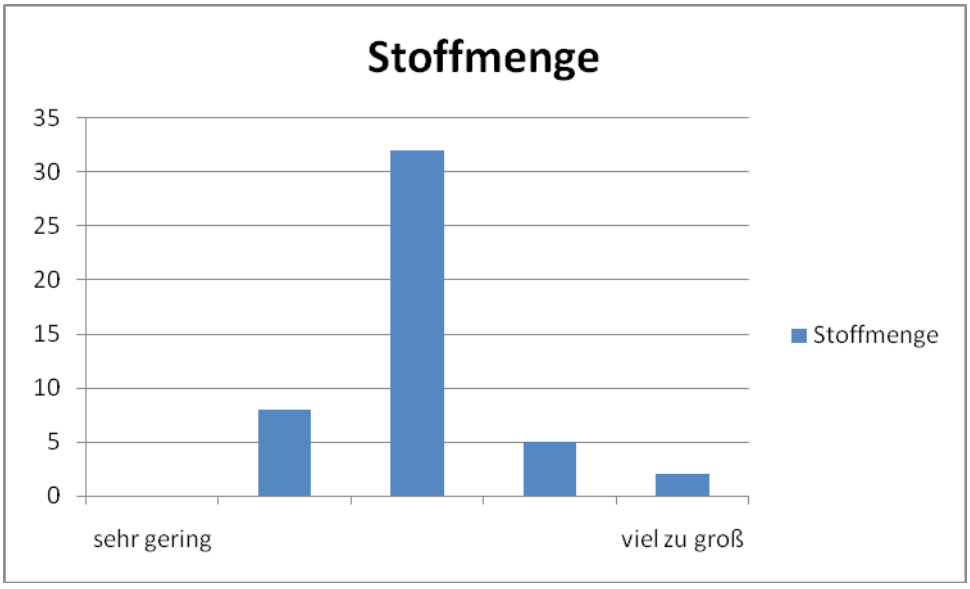

Nationalität
50
45
40
35
30
25
20
15
10
5
0
Deutsch
UK
NA
Nationalität

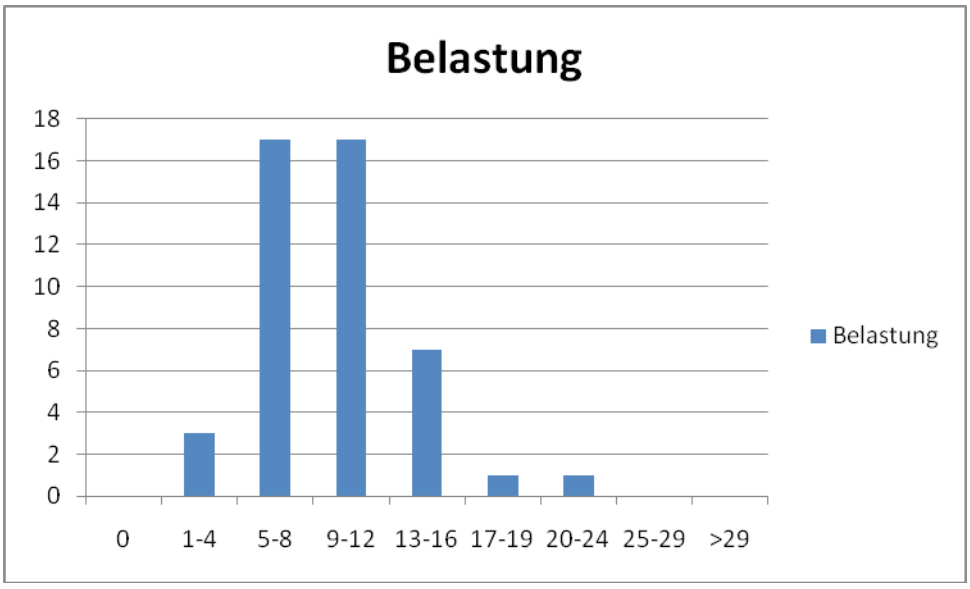

Leistung
25
20
15
10
5
0
sehr schlecht
sehr gut
Leistung

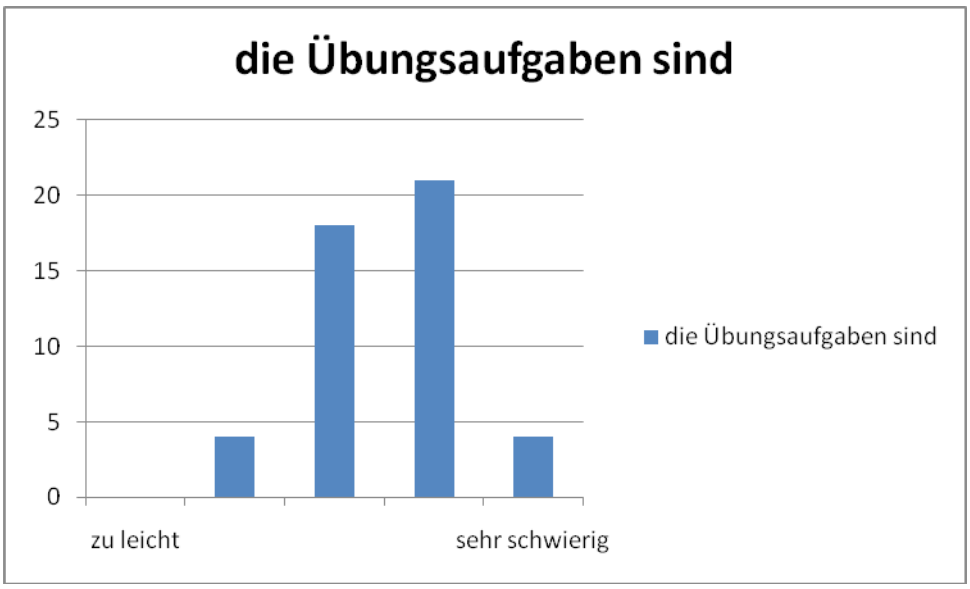

Lehrstoff
25
20
15
10
5
0
sehr schlecht
sehr gut
Lehrstoff

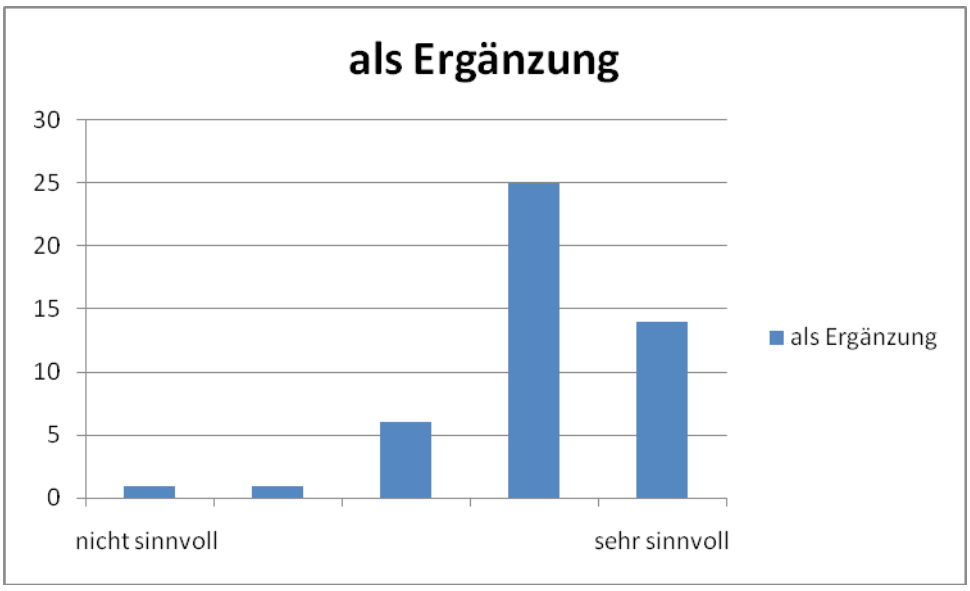

Begleitmaterial
30
25
20
15
10
5
0
gar nicht hilfreich
sehr hilfreich
Begleitmaterial

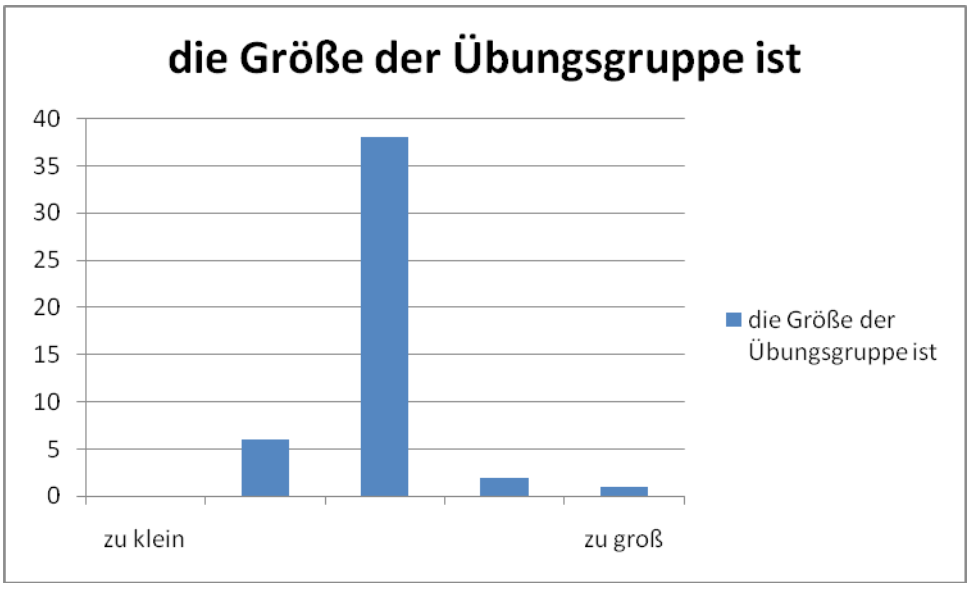

Veranstaltung
35
30
25
20
15
10
5
0
unterfordert
überfordert
Veranstaltung


Stoffmenge
35
30
25
20
15
10
5
0
sehr gering
viel zu groß
Stoffmenge


Belastung
18
16
14
12
10
8
6
4
2
0
0
1-4
5-8
9-12
13-16
17-19
20-24
25-29
>29
Belastung


die Übungsaufgaben sind
25
20
15
10
5
0
zu leicht
sehr schwierig
die Übungsaufgaben sind


als Ergänzung
30
25
20
15
10
5
0
nicht sinnvoll
sehr sinnvoll
als Ergänzung


die Größe der Übungsgruppe ist
40
35
30
25
20
15
10
5
0
zu klein
zu groß
die Größe der Übungsgruppe ist

14

Antworten

Deutsch	UK	NA
45	1	1

47

BSc. 1FS	Sc. (Mathe) 1F	BSc. NA	SHF 1FS	Dip (Bio) 9FS
39	1	3	4	2

47

Deutsch
47

sehr schlecht				sehr gut
1	1	3	22	20
2%	2%	6%	47%	43%

47

sehr schlecht				sehr gut
0	3	7	16	21
0%	6%	15%	34%	45%

47

gar nicht hilfreich				sehr hilfreich
1	1	4	27	14
2%	2%	9%	57%	30%

47

unterfordert				überfordert	
1	12	29	4	1	47
2%	26%	62%	9%	2%	

sehr gering				viel zu groß	
0	8	32	5	2	47
0%	17%	68%	11%	4%	

0	1-4	5-8	9-12	13-16	17-19	20-24	25-29	>29	
0	3	17	17	7	1	1	0	0	46
0%	7%	37%	37%	15%	2%	2%	0%	0%	

zu leicht				sehr schwierig	
0	4	18	21	4	47
0%	9%	38%	45%	9%	

nicht sinnvoll				sehr sinnvoll	
1	1	6	25	14	47
2%	2%	13%	53%	30%	

sehr schlecht verständlich				sehr gut verständlich

ors?

zu klein				zu groß	
0	6	38	2	1	47
0%	13%	81%	4%	2%	

Informatik 1
PD Dr. Jan-Georg Smaus
Wintersemester 2010/11

Freitext Kommentare

2.8

- Motivation des Dozenten. Ausführliche Vorlesungsunterlagen. Konsequente Umsetzung. Hintergrundinformation (mehr wäre auch schön). Interaktion mit Studierenden.
- Betreutes Programmieren.
- Lernen von mehreren Sprachen. Auch manchmal lustig.
- Betreutes Programmieren. Alle Vorlesungsmaterialen online. Vorlesung gut verständlich (Inhalt und Form).
- Python.
- Python. Aufzeichnungen.
- Nur jede zweite Woche ein Übungsblatt.
- Richtig “dosierte” Stoffmenge. Gute Vorlesungen. Die Beweise sind machbar.
- Der Dozent ist sehr kompetent und versucht häufig Beispiele “aus dem Leben” zu finden.
- Python und die schnellere Vorgehensweise bei dieser Sprache.
- Aufzeichnung; konkrete Beispiele; Schritt für Schritt Erklärung.
- Das die Vorlesungen aufgezeichnet werden.
- Lecturnity Aufzeichnungen zur nachbereiten. Python als Programmiersprache.
- Aufzeichnung. Struktur. Erklärungen.
- Beispielhafte Erklärung. Möglichkeit zum Nachhören der Vorlesung.
- Klarer Aufbau und ausführliche Herangehensweise.
- Ausführliche Erklärungen.
- Für mich persönlich viel neues bei gutem Lehrtempo.
- Gute Aufteilung von Praxis und Theorie. Angenehmer Wechsel von Scheme zu Python.
- Sehr schöne mit Latex erstellte Folien. 2 Programmiersprachen wurden vorgestellt.
- Ich bekomme genau das vermittelt, was ich mir vorgestellt habe und darüber hinaus noch viel mehr!
- Strukturierung eingängig. Durch die Aufzeichnung kann der Vorlesungsstoff jederzeit ohne Einschränkungen nachgearbeitet werden; man muss während der Vorlesung nicht mitschreiben, kann sich gut auf die Vorlesung konzentrieren! Es macht sehr viel Spaß Hr. Smaus zuzuhören, perfekte Ansprechhaltung! Selten, so eine gute Vorlesung besucht!
- Scheme, Python → interessante Programmsprachen und Vergleich zwischen funktionellen und imperative Programmsprachen. Klarer Aufbau und Strukturierung, Probeklausur, 4in1Ausdruck.
- Python, viele “gute” Beispiel, electures, BP, Atmosphäre; gute Einstiege in neue Kapitel durch clever gewählte Beispiel, Probeklausur, Dozent.
- 2 Programmiersprachen (Scheme / Python). Betreutes Programmieren. Sehr gut verständliche Erklärungen des Dozenten.
- Gut verständlich. Strukturiert. Gute Unterstützung durch betreutes Programmieren.
- Sehr gute Einführung in beide Programmiersprachen. Bereitschaft ausgiebig auf Fragen einzugehen.
- Mir gefallen die gelegentlichen humorvollen Einlagen in der Vorlesung, die sprachliche Nähe von Professor zum Student.

- Vortragstil.
- Gut nachvollziehbare Vermittlung des Stoffes und Einführung in die Programmierung.
- Wechsel von Scheme zu Python.
- Es werden Codebeispiele vorgestellt. Es werden 2 Sprachen (Scheme und Python) behandelt, die sich stark unterscheiden!! Sehr gut.
- Das Grundkonzept funktionaler Programmierung wurde greifbar vermittelt.
- Das betreute Programmieren zusätzlich zu den Übungen. Python. Kleinere Gags bringen Abwechslung.
- Dozent ist gut gelaunt. Nicht nur Scheme wird gemacht.
- Python
- Die ruhige Art und Weise, wie der Dozent vorträgt.
- Das wir Scheme abgeschlossen haben.
- Es werden immer Code Beispiele zu jedem Thema gezeigt.
- Schön übersichtliche Folien, die auch zeitnah online gestellt werden. Nicht zu viel Stoff. Ausführliche Erklärungen zu Prozeduren / Funktionen etc..
- Aufzeichnung. Veranschaulichung. Motivation. Gut verständlich.

2.9

- Einschränkung beim BP über DMDA. “Dumm denken” Aufgaben im BP. Keine .vimrc oder .emacs bei BP. Keine Syntax highlighting in vim verfügbar.
- Bisher sehr Scheme-lastig.
- Gelegentlich zu ausführlich , vielfache Wiederholung.
- Scheme.
- Scheme.
- Es wird viel nur abgelesen vom Dozenten. Oft laut in Vorlesung.
- Scheme. Es werden manchmal unwesentliche Themen genauer behandelt, als wesentliche.
- Scheme, Schokoladenkeks.
- Wenig anschaulich zu theoretisch. Chocolate cookie.
- Zu einfache Beispiele, die nicht immer aufschlussgeben, wie man sie verwenden könnte.
- Bisschen unübersichtlich, komische Beispiele.
- Scheme als Programmiersprache.
- Theorie Sachen könnten anschaulicher sein.
- Schnelles Tempo. Extrem viel Stoff auf einmal. Druck der 50 % Marke zu Zulassung zur Klausur.
- Der Blick fürs Wesentliche: An manchen Stellen zu viel Begründung, warum das Thema so gelehrt wird. Diese Zeit fehlt bei Erklärungen bei wichtigen Themen. (z.B. “ Türme von Hanoi”)
- Bei einfachen Sachen wird manchmal zu ausführlich erklärt und dagegen über schwierige Sachen manchmal hinweggegangen.
- Nicht genügend Tiefe.
- Stoff wird etwas statisch vermittelt, manchmal fehlt eine Motivation (nicht meine persönliche, sondern eine, die den Weg zur Lösung eines Problems erklärt). Mehr Dialog, wo möglich.
- Scheme, Theorieübungsblätter teilweise deutlich schwieriger als Vorlesung / BP / Folien/ …
- Struktur (Scheme ohne Lambda-Kalkül). Geringe theoretische Tiefe. Sehr strikte Programmiervorgaben, wenig Möglichkeiten einen eigenen Stil zu entwickeln. “Konstruktionsvorlagen”!

- Zu viel Wiederholung. Inhalt des ersten Teils (Scheme – Programmierung) ist praxisfern.
- Tempo teilweise zu niedrig.
- Unausgeglichene Prioritäten der Vermittlungen von Informationen; weniger wichtiges bzw. Leicht verständliche Dinge werden oft zu lange und ausführlich betrachtet, so dass wichtigere / schwerere Themen oft zu kurz kommen.
- Manchmal vielleicht etwas zu umständlich formulierte Erklärungen.
- Manchmal vielleicht zu viel Wiederholgen in den folgenden Stunden.
- Einige Dinge werden zu oft wiederholt, obwohl sie trivial sind.
- Die Aufgaben bei BP und zuhause sind teilweise ein wenig aus der Luft gegriffen. Anders gesagt: Es werden mitunter abstruse Aufgabenkonstrukte gestellt, deren Zweck schleierhaft ist und die deshalb unnötig schwer verständlich sind.
- Scheme
- Erklärungen fehlen.
- Die Kapazität des W-LAN Netzes.
- Scheme.
- Scheme.
- Das Niveau der Vorlesung unterscheidet sich stark vom Niveau der Übungen (Übungen sind um einiges schwieriger). Das Tempo der Vorlesung ist sehr schleppend.
- Manchmal etwas holperig.

2.10

- Was wurde aus den Fleißaufgaben?
- Weniger Scheme.
- Weniger Scheme.
- Hochladen der annotierten Folien.
- Scheme eignet sich nicht als Einstieg für Leute, die schon Kontakt mit anderen Programmiersprachen hatten z.B. C++ oder Delphi.
- Scheme abschaffen; schneller das Problem auf den Punkt bringen; Mehr Python.
- Sinnvollere Beispiele.
- Sie sollte mehr Gewicht haben. 3. Programmiersprache.
- Das betreute Programmieren bringt mir persönlich nichts. Meiner Meinung nach sollte die Teilnahme freiwillig sein. Dafür könnte man auch den Schwierigkeitsgrad der Aufgaben erhöhen, da man mit mehr Zeit als nur wöchentlich 2 Stunden kompliziertere Probleme behandeln könnte.
- Ab und zu wünsche ich mir alternative (eher theoretische) Erklärungen zu Konzepten (also z.B. Nicht anhand von Scheme die Umsetzung zeigen, sondern das zugrundeliegende Konzept erläutern (zusätzlich!))
- Eventuell mehr Übungen auf der Homepage bereitstellen. Es sollten unbedingt vor der Klausur (oder Nachschriebklausur) zusätzliche Termine bereitgestellt werden, in denen man im PC-Pool die Übungsblätter zum BP noch einmal durch programmieren kann. Das wäre für den Lernerfolg großartig und würde mich und viele andere Kommilitonen gut auf die Klausur vorbereiten.
- Eventuell Skript (als Band) vor den Veranstaltungen verteilen → siehe MST. Mehr Übungen (unterschiedliche Schwierigkeitsstufen).
- BP(Theorieblätter/Probeklausur nicht nur als Zulassungskriterium, sondern überschüssige Punkte als Bonus für Endklausur aufrechnen! (→ Motivation)
- Etwas höheres Tempo. Weniger Didaktik, mehr Inhalt.
- “Vorlesung” wörtlicher nehmen! Auf Interaktion der Hörer muss kein Wert gelegt werden!

Lesen → Hören

- Ein zusammengefasstes Skript zusätzlich zu den Folien wäre noch toll. Weniger Dr.Rakket und mehr Python. Bonuspunkte einführen (meiner Meinung nach der besten Motivation zum Lernen).
- Eventuell mehr Beispiel vor Ort programmieren.
- Keine, gut so wie es ist.
- Wir behandeln in einem Semester Scheme und Python unter dem Thema “funktionale Programmierung”. Als Anhänger dieses Konzeptes finde ich es besser eine praktisch verwendbare funktionale Sprache zu lernen (Haskell) und darin professionell zu werden, als in zwei Sprachen mit letztlich sehr unterschiedlichen Konzepten oberflächlich zu bleiben.
- Python vor Weihnachten!
- Life of Brian komplett gucken bei der Einführung in Python.
- Falls Scheme weiterhin gelehrt werden soll, vielleicht weniger davon, aber mehr Python. Python ist besser verständlich, auch für mich, der bis jetzt nur Scheme und Python kennt.
- Andere Programmiersprache als Scheme behandeln.
- Vorlesung an die Übungen anpassen bzw. umgekehrt. Mehr praktische Beispiele / Anwendungen.
- Am Anfang nicht so schnell, für die schwächeren Studenten.

3.5

Informatik 1
PD Dr. Jan-Georg Smaus
Wintersemester 2010/11

Lehrpreis Kommentare

- Nette Persönlichkeit, sehr freundlich, geht sofort auf Meldungen ein, kommt den Studenten sehr entgegen, lässt mit sich reden und die Vorlesungen sind gut gestaltet.
- Da ich nur zwischen 4 unterschiedlichen Vorlesungen entscheiden kann und dies dabei die beste ist.
- Im Vergleich zu allen anderen Vorlesungen, hält Dr. Smaus seine Vorlesung intuitiv am besten, möglicherweise auch der Vergleich der Komplexität anderer Fächer.
- Hohe Motivation, gibt sich sehr viel Mühe den Stoff allen verständlich zu erklären.
- Dozent Smaus engagiert sich sehr stark und man merkt, dass er sehr, sehr, sehr viel für die Vorlesungen vorbereitet. Vermutlich sogar viel Freizeit dafür opfert!! Hut ab!
- Guter Einstieg in Informatik, klare und strukturierte Vorlesung, gute Skripte (auch 4in1 Ausdruck)
- Super Ansprechhaltung, es macht sehr viel Freude der Vorlesung zu folgen. Vorlesung super strukturiert, interessante Nebeninformationen (z.B. Spam), Dozent absolut bemüht und freundlich.
- Stylischer Bart.
- Er kann sich besonders gut in uns hineinversetzen.
- Besonders anfängerfreundliche Vorlesung.
- Sehr gute Vorlesung für Erstsemester aufgebaut, Vorlesungen werden aufgezeichnet und man kann auch in den Büchern nachlesen.
- NN
- Hervorragende Vorlesungen die vom Schwierigkeitsgrad genau richtig sind!
- Motivation!