



FAU • Dekanat der TF
Martensstraße 5a
91058 Erlangen

FAU • Dekanat der TF • Martensstr. 5a • 91058 Erlangen

Herr
M. Sc. Simon Ruderich
(PERSÖNLICH)

SS'19: Auswertung zu Übungen zu Systemprogrammierung 1

Sehr geehrter Herr M. Sc. Ruderich,

Sie erhalten hier die Ergebnisse der automatisierten Auswertung der Lehrveranstaltungsevaluation im SS'19 zu Ihrer Umfrage vom Typ "Übung":

- Übungen zu Systemprogrammierung 1 -

Es wurde hierbei der Fragebogen - t_s19u29 - verwendet, es wurden 87 Fragebögen von Studierenden ausgefüllt.

Der Wert 1 kennzeichnet hierbei eine maximale Güte, der Wert 5 eine minimale Güte für die einzelnen Fragen bzw. Mittelwerte.

Der Kapitel-Indikator für "3. Hauptfragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter" zeigt den mit der Anzahl der Antworten gewichteten Mittelwert der 6 Hauptfragen und damit den Lehrqualitätsindex (LQI), dieser wird bei genügend (ab 5) Rückläufern zur Qualitätssicherung durch die Studienkommissionen und auch für die Bestenlisten der verschiedenen Kategorien verwendet.

Der Kapitel-Indikator für "5. Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter" zeigt den Mittelwert für die restlichen Einzelfragen, diese dienen nur der Information der Dozentin/des Dozenten.

Bei den Einzelfragen werden je nach Fragen-Typ die Anzahl und Verteilung der Antworten, Mittelwert und Standardabweichung aufgelistet.

Die Text-Antworten für jede offene Frage sind zusammengefasst aufgelistet.

Eine Profillinie zeigt den Vergleich zu den Mittelwerten aller Rückläufer für diesen Fragebogen-Typ (wird extra per E-Mail zugesandt). Die Profillinie eignet sich auch zur Präsentation in der LV.

Eine Einordnung Ihrer Bewertung ist nach Abschluss der Ergebnisauswertung unter

<http://eva.tf.fau.de> --> Ergebnisse --> SS'19 möglich, siehe Bestenlisten, Percentile, etc.

Bitte melden Sie an tf-evaluation@fau.de die Anzahl der ausgegebenen TANn, wenn Sie das bis jetzt versäumt haben.

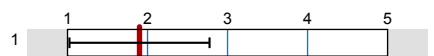
Mit freundlichen Grüßen

Kai Willner (Studiendekan, kai.willner@fau.de)
Jürgen Frickel (Evaluationskoordinator, tf-evaluation@fau.de)



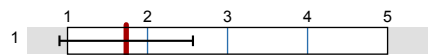
Globalwerte

3. Hauptfragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter



mw=1,9
s=0,88

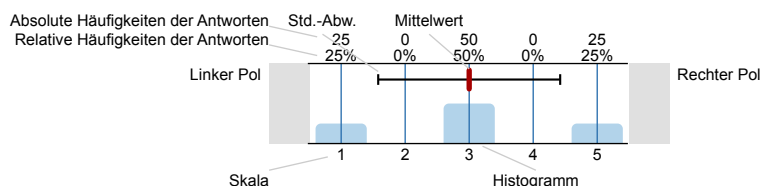
5. Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter



mw=1,73
s=0,83

Legende

Fragetext



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

2. Allgemeines zur Person und zur Lehrveranstaltung

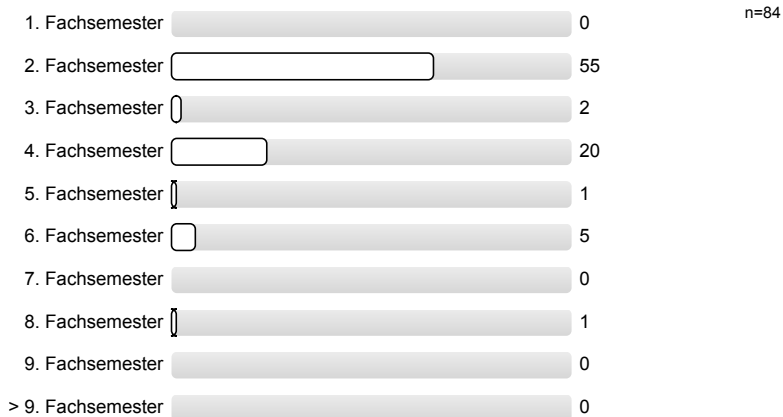
2.1) ▶▶ Ich studiere folgenden Studiengang:

CE • Computational Engineering	<input type="text" value="9"/>	9	n=87
EEI • Elektrotechnik - Elektronik - Informationstechnik	<input type="text" value="1"/>	1	
INF • Informatik	<input type="text" value="47"/>	47	
INFLA • Informatik für Lehramt	<input type="text" value="2"/>	2	
IuK • Informations- und Kommunikationstechnik	<input type="text" value="7"/>	7	
MT • Medizintechnik	<input type="text" value="12"/>	12	
TMath • Technomathematik	<input type="text" value="1"/>	1	
WINF • Wirtschaftsinformatik	<input type="text" value="6"/>	6	
Sonstiges	<input type="text" value="2"/>	2	

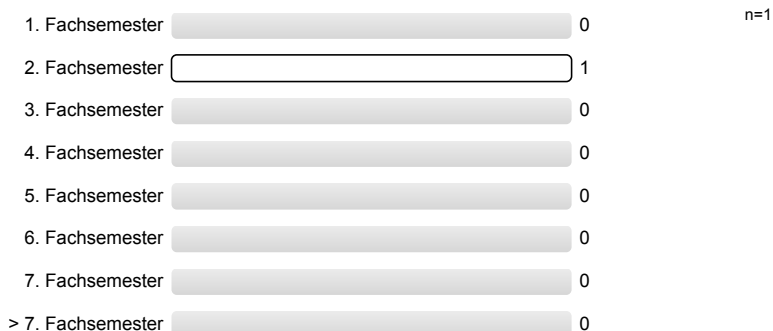
2.2) ▶▶ Ich mache folgenden Abschluss:

B.Sc. • Bachelor of Science	<input type="text" value="83"/>	83	n=87
M.Sc. • Master of Science	<input type="text" value="1"/>	1	
M.Sc.(hons) • Master of Science with Honours	<input type="text" value="0"/>	0	
M.Ed. • Master of Education	<input type="text" value="0"/>	0	
LA • Lehramt mit Staatsexamen	<input type="text" value="2"/>	2	
Dr.-Ing. • Promotion	<input type="text" value="0"/>	0	
Zwei-Fach-Bachelor of Arts	<input type="text" value="1"/>	1	
Sonstiges	<input type="text" value="0"/>	0	

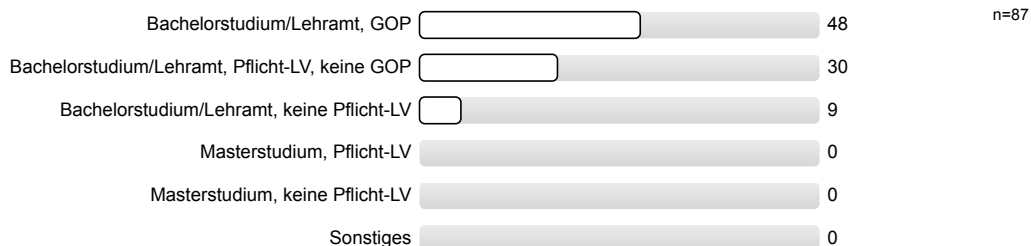
2.3) Ich bin im folgenden Fachsemester (im Bachelor):



2.4) Ich bin im folgenden Fachsemester (im Master):



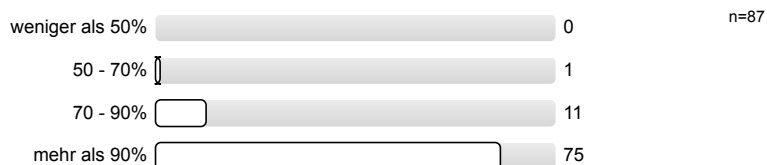
2.5) ►► Diese Lehrveranstaltung gehört für mich zum



2.6) Als Studiengang bzw. Abschluss ist *Sonstiges* ausgewählt, ich studiere folgende Kombination:

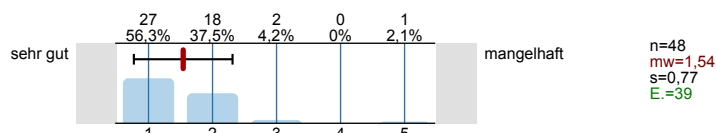
- Mathematik
- Mathematik

2.7) Ich besuche etwa Prozent dieser Übung.

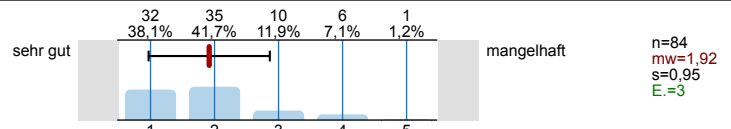


3. Hauptfragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter

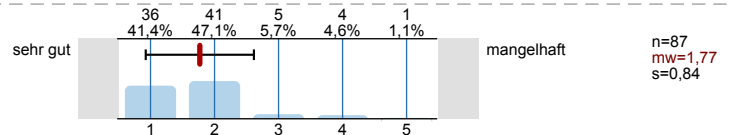
3.1) ►► Die Übung entspricht den im Modulhandbuch eingetragenen Inhalten und Kompetenzen.



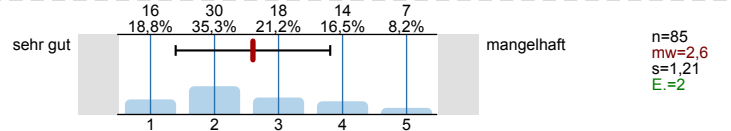
3.2) ►► Wie ist die Einpassung in den Studienverlauf Ihres Studienganges?



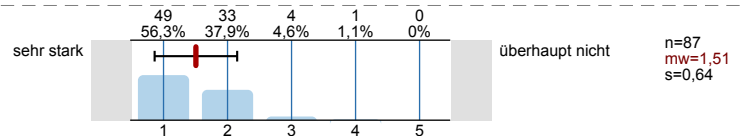
3.3) ►► Wie ist die Übung selbst strukturiert?



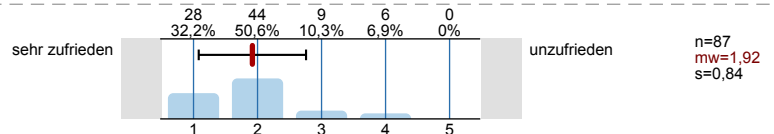
3.4) ►► Wie ist die Übung inhaltlich und organisatorisch mit der zugehörigen Vorlesung abgestimmt?



3.5) ►► Die Übungsleiterin/Der Übungsleiter wirkt engagiert und motiviert bei der Durchführung der Übung.



3.6) ►► Wie zufrieden sind Sie insgesamt mit der Übung:



4. Kommentare zu Lehrveranstaltung und Übungsleiterin/Übungsleiter

4.1) An der Lehrveranstaltung gefallen mir folgende Aspekte besonders:

- -Folien sind sehr gut
- -Milan war immer freundlich und hilfsbereit, kann echt gut erklären und man merkt, dass er sehr viel Ahnung hat, von was er da spricht
- die meisten Tutoren sind in der Rechnerübung sehr geduldig und hilfsbereit
- 95% der Tutoren in den Rechnerübungen sind super engagiert und nett, habe extrem viel in den Übungen gelernt
Tafelübung bei Milan und Felix waren super
- A.Feiler
- Erklärungen
- Fokus auf das Wichtige
- Bei Anna ist immer eine angenehme und lockere Atmosphäre, was sehr angenehm ist. Fragen konnte sie immer gut und verständlich beantworten und auch auf Emails hat sie schnell geantwortet. Ich finde sie kann sehr gut erklären und durch ihren Vortragsstil kommen die Zusammenhänge viel besser raus, als bspw in den Folien.
- Besonders gefallen hat mir unsere Tutorin Eva Dengler. Du bist sehr engagiert und gibst dir sehr viel Mühe uns den Stoff zu erklären. Du bist auch um 8 Uhr in der Früh sehr motiviert und lustig. Du machst das wirklich sehr gut. Mach doch nächstes Semester bitte SP2 Tutorin :)
(Hier wäre eine Katze, wenn es sie richtig formatieren würde)
- Das die Abgabe der Aufgaben abhängig von der Tafelübung und nicht des Veröffentlichungsdatums sind
- Dass immer eine Programmieraufgabe zusammen gelöst wird, die mit der Aufgabe, die man selbst programmieren soll, zusammenhängt
- Der Einblick in die kommenden Übungen
- Der Gruppenbonus ist eine gute Idee
- Di T11:
- Tutor war sehr engagiert (auch mit der eigenen Website mit Hilfestellungen) was sehr geholfen hat
- Die Hausaufgaben sind anspruchsvoll und zeitaufwendig, aber dennoch sehr lehrreich und machen durchaus Spaß. Sie vermitteln die verschiedenen Konzepte der angewandten Systemprogrammierung und die damit einhergehenden Schwierigkeiten und Besonderheiten in der Programmierung sehr gut.

Die Übung ist in ihrem Aufbau und in den zur Verfügung gestellten Materialien sowie mit dem SVN-Abgabesystem gut organisiert, auch wenn das Abgabeframework anfangs etwas schwer durchschaubar war. Die Übungsseite und speziell die dortigen Korrekturhinweise sind gute Orientierungspunkte für das richtige Lösen der Aufgaben.

Der Tutor Lou Knauer hält seine Übung (trotz der meist sehr geringen Teilnehmerzahl) mit viel Motivation und Enthusiasmus und man merkt, dass er viel vom Thema versteht und sich dafür begeistern kann. Er erklärt die behandelten Konzepte und Aufgabenstellungen gut, beantwortet Fragen sehr kompetent und bietet dazu auch viel Hintergrundwissen.

Seine Korrekturen sind gut verständlich und fair, und er steht auch stets für Nachfragen zur Verfügung. Kann ich weiterempfehlen, 10/10.

- Die Korrektur der Übungsaufgaben per Hand von einem Menschen finde ich super, da man so wenigstens erfährt, was man alles falsch gemacht hat, bzw. was man alles besser/schöner machen könnte.
- Die Themen sind interessant und C Programmierung ist für mich eine neue Art und Weise zu Programmieren. Die Aufgaben in Gruppenarbeit zu lösen finde ich hier sehr sinnvoll.
- Die klare Struktur der Übungsfolien
- Die starke Praxisorientierung mit richtigen Programmier-Aufgaben.
- Die Übung und deren Inhalte sind gut gewählt!
- Die Übung war m. M. n. gut strukturiert und alles wurde sehr verständlich erklärt. Die SP Übungen haben mir bisher in den ersten 2 FS am meisten Spaß gemacht.
- Es ist schon angenehm anstrengend
- Gute Übung bei Ruben, sehr zufrieden.
- Guter Einstieg in die Programmierung mit C.
- Hier wird der Inhalt zu den Übungsaufgaben klarer als in der Vorlesung
- Interaktivität, Anwendungsnahe Konzepte im Mittelpunkt
- Julian Preis: Der motivierte Tutor^^
- Julian ist sehr motiviert, immer pünktlich und freundlich und nimmt sich auch immer zeit, Leuten die nicht Informatik studieren, Dinge ausführlicher zu erklären. 10/10 würde wieder besuchen, falls ich durchfalle, was ich natürlich nicht werde, da ich so gut vorbereitet wurde
- Man bekommt praktisch ein viel besseres Verständnis dafür, was genau im Hintergrund bei der Programmausführung passiert. Die Anzahl der angebotenen Übungen ist super.
- Ruben Baecker erklärt alles ganz gut, vor allem die Make Datei
- Segmentation fault (core dumped) (° °)
- T08: es wird sich immer genug Zeit für Fragen genommen und Inhalte bei Bedarf wiederholt oder ausführlicher besprochen
- T23: Tutor ist kompetent
- Tutor Kay(Do 8Uhr) erklärt sehr ausführlich und genau und geht dabei auch zwischen drin auf Fragen ein, bis das Thema jedem soweit verständlich ist.
- Zu Knauer, Lou (T07): tolle Tafelübung mit gutem Verhältnis aus Abfragen und Vorstellung der neuen Inhalte. Trotz mäßiger Anwesenheitsquote wirkte er motiviert und interessierte sich auch für unsere Fragen. Würde auch SP2 bei ihm besuchen wollen.
- praktische Anwendung C
- Übung Kay Friedrich: Dass am Schluss immer noch ein Beispiel gemacht wird, dass bei der Hausaufgabe hilfreich sein kann.
- Übungsleiter versteht sein Fach sehr gut.

4.2) An der Lehrveranstaltung gefällt mir Folgendes weniger, und ich schlage zur Verbesserung vor:

- - Anwesenheitspflicht ist nicht unbedingt sinnvoll
- die Bewertung der Übungen ist teilweise trotz der Korrekturhinweise unfair, bzw. weicht von ihnen ab
- Manche dürfen nach Abgabeschluss noch Hausaufgaben abgeben
- Stefan schweift manchmal sehr stark vom eigentlichen Thema ab
- - Ich habe SP jetzt nachgeholt und tue mir im 6. Semester wesentlich leichter als ich es damals im 2. fand (wahrsch Wissen aus anderen Veranstaltungen) - evtl. ist SP zu früh?
- evtl. auch Theorieaufgaben in die Übungen bringen.
- - Simon Ruderich ist in der Rechnerübung ziemlich von oben herab auf die Fragen eingegangen. Gerade wenn man noch nie vorher in C programmiert hat und schon froh ist überhaupt was zustande gebracht zu haben, trägt das überhaupt nicht zu Motivation bei. Ich weiß ja, dass ich wenig Ahnung von SP habe, aber genau deswegen bin ich ja da...
- manche entscheidende (z.B. poison pill) wurden weder in der VL noch in der Übung oder auf dem Aufgabenblatt erwähnt, waren aber trotzdem essentiell zur Lösung der Aufgaben

- A. Feiler
nix
- Als Verbesserung schlage ich vor, mehr Programmierbeispiele in die Übungsfolien einzubauen, um die Verwendung von bestimmten Funktionen besser darzustellen
- Also ich finde es ja gut wenn live in der Übung gecoded wird.
Aber ne Schwarze Shell kann man echt nicht gut auf den Beamern lesen.
Oft wird das mit: "Das kann ich grad nicht besser einstellen" abgetan:
In der Berufsschule haben wir Code immer mit Schwarzer Schrift auf weißen Grund präsentiert, das ist auf jeden beamer gut lesbar ;)
- Anwesenheitspflicht, oft sind die Hausaufgaben zu Zeitaufwändig und die meiste Zeit geht für langweilige Fehlerbehandlung drauf
- Auf richtige Fehlerbehandlung könnte am Anfang etwas näher eingegangen werden, dass war bei der zweiten Hausaufgabe ziemlich ernüchternd.
- Bepunktungssystem steht nicht im Verhältnis zum Aufwand. Wenn man ein Programm schreibt, dass zu 100% mit der Ausgabe des Lehrstuhl-Programms übereinstimmt, und man z.B. eine Fehlerbehandlung vergisst, bekommt man nur die Hälfte der Punkte.
Desweiteren wird für den selben Fehler (z.B. bei einer Fkt auf den falschen Parameter bei der Fehlerbehandlung getestet) immer wieder 1 Punkt abgezogen, was insgesamt zu 4 Punkten Abzug geführt hat. Unnötig und unfair!!!
Aussagen von versch. Tutoren widersprechen sich. z.B. wurde zu Beginn des Semesters gesagt, printf muss nicht fehlerbehandelt werden, was ich dann auch bei Aufgabe 1-5 nicht gemacht habe und plötzlich bekomme ich 3 Punkte abgezogen weil in Aufgabe 5 printf nicht fehlerbehandelt wurde.
- Dadurch, dass die Rechnerübungen für SP 1 und SP 2 waren, haben sich einige Tutoren nicht für die SP1 Übungen zuständig gefühlt bzw. kannten die Aufgaben nicht. Vielleicht ist es hilfreich, zu kennzeichnen, welche Übung eher für SP1 und welche eher für SP2 gedacht ist, je nachdem, welche Tutoren gerade da sind. Grundsätzlich ist die hohe Anzahl an Übungen super! Diese werden auch gebraucht, da die Abgaben nicht wirklich alleine zuhause zu erledigen sind.
- Das Abgabesystem finde ich unnötig kompliziert, man könnte stattdessen ein wesentlich einfacheres System implementieren, da die Versionsverwaltung größtenteils sowieso unnötig ist, weil man die Aufgaben zumeist sowieso gleichzeitig bearbeitet oder sich untereinander abspricht und nur die letzte Version hochlädt.
Eine kurze Übersicht über alle Methoden, die man benutzt, wäre manchmal sinnvoll
- Der fehlende Zusammenhang zur Vorlesung
- Die Anforderungen der Aufgaben sind nicht erkennbar. Man kann trotz funktionierendem Programm 0 Punkte erhalten. Ein rein Fehlerbasiertes Korrektursystem regt nicht zur Motivation bei folgenden Übungen an.
- Die Folien geben teilweise sehr wenige Informationen zu Themen und Funktionen, steht zwar alles in der Man-Page aber könnte man ja trotzdem ein bisschen ausführlicher hinschreiben. Ausserdem wäre ein Blatt gut auf welchem alles zu SVN nochmal zusammengefasst ist
- Die Präsentationen der Kommilitonen sind manchmal unvollständig und oder es wurden nicht immer optimale Lösungen gewählt.
Sinnvoller wäre die Präsentation durch den Übungsleiter.
- Die Veranstaltung hat keinen erkennbaren Bezug zu der Vorlesung. Hier wären Verbesserungen notwendig zwischen den inhaltlichen Abstimmung zwischen Vorlesung und Übung.

Des Weiteren sind in den Rechnerübung viele Tutoren nicht besonders motiviert. Eine neue Programmiersprache und die Feinheiten der Aufgaben zu verstehen erfordert seine Zeit. Tutoren die einem nur das Gefühl hinterlassen, dass man dumm ist weil man etwas nicht sofort versteht fördern nicht die Motivation für das Fach. Teilweise wussten die Tutoren selber nicht wie die Aufgabenstellung zu verstehen und man hat dann von verschiedenen Tutoren unterschiedliche Aussagen zu der gleiche Sache erhalten. Da muss man dann selber entscheiden wem man glauben kann und wem nicht. Das ist für mich keine anständige Vorbereitung um dann in Rechnerübungen die Studierenden zu helfen. Die Tutoren müssen verstehen was sie erklären, sonst kann man sich die Rechnerübungen auch gleich schenken.
- Die Verwendung von SVN statt GIT als Versions-Management-Tool halte ich für veraltet. Da GIT deutlich beliebter und weiter verbreitet ist, wäre das ein Anreiz für mich gewesen, tatsächlich Versions-Management für meinen Code zu verwenden, statt das ganze nur als ein Abgabe-Tool zu sehen.
- Die verschiedenen Übungsleiter korrigieren trotz der (wirklich sehr hilfreichen) neuen Korrekturhinweisen sehr unterschiedlich
- Die versprochenen Mails mit Materialien, die wir in den Übungen besprochen haben, wurden leider im Nachhinein fast nie geschickt
- Die Übung hat nichts mit der Vorlesung zu tun, man kann für die Übungen nichts nochmal nachschlagen, weshalb man sich so fühlt als wäre Übung und Vorlesung zwei einzelne Veranstaltungen! In der Vorlesung bitt Code-Beispiele wie in AuD/PfP/Gra machen, in den Übungen mehr kleine Aufgaben machen um einzelne befehle bessere zu erklären und nicht das ganze Programm in einer Stunde
- Die Übungen sind extrem zeitaufwendig, finde da wäre ein Übungsschein angemessen

Korrektur teilweise sehr streng, zumindest bei Folgefehlern/mehrmals auftretenden Fehlern weniger abziehen

Est statt Waffel nutzen
- Die Übungsinhalte und -Aufgaben sind leider meist nur schlecht mit der Vorlesung abgestimmt.

Für unerfahrene Programmierer ist der rapide Einstieg in C teilweise schwierig, da die Sprache nur kurz innerhalb von 2-3 Wochen umrissen wird und sich vom Konzept doch stark von vorher behandelten Sprachen (e.g. Java) unterscheidet; dadurch hat man manchmal mehr Probleme mit der Sprache als mit der eigentlichen Aufgabenstellung.

- Einsatz oder zumindest Hinweisen auf moderne Tools zur C-Entwicklung (etwa CLion von JetBrains) --> moderneres Auftreten striktes Vorgehen bei Fehlerprüfung verständlich, jedoch an einigen Stellen übertrieben (vielleicht eher gegen Linux API entwickeln statt POSIX)
Fehlte Information darüber, dass primär POSIX Pages gelten (vielleicht in Materialien, ... statt printf(3) printf(3p) schreiben würde Anforderungen klarer machen
- Es sollten weniger Punkte bei den Aufgaben abgezogen werden bzw. eine mindest Punktzahl für funktionierende Abgaben geben
- Es wäre nett wenn deutlich gemacht wird was aus der Vorlesung die Übung vertieft. Maßgeblich scheint mir die Übung Programmiertraining in C zu sein, während die Vorlesung Inhalte vermitteln will.
- Kein Zusammenhang zwischen der Übung (C-Programmierung) und der Vorlesung.
- Lächerliches Bewertungssystem der Übungen. Meist keine Kommunikation des Bewertungsrahmens, sogar Rückwirkende Änderungen des Bewertungshorizontes.
Aufgabenstellungen sind meist so schwammig, dass man anhand der (manmal fehlerhaften) Referenzimplementierung, herausfinden muss was gefordert ist.
Fehlerabfrage für jede Zeile Code ist m.E. nicht unbedingt sinnvoll
- Manchmal gibt es relativ viel Abzug von Bonuspunkten. Wäre super, wenn das zumindest ein BISSCHEN besser bewertet wird zur besseren Motivation.
- Nicht hilfreiche Kommentare der Tutoren einfach sein lassen
- Nicht so strenge Bewertung, da es bei derart zeitaufwendigen Abgaben (von allen Modulen des 2. Semesters größter Zeitaufwand!) demotivierend ist, kaum Punkte zu erhalten. Das nimmt die Motivation weiter mitzuarbeiten, obwohl C eigentlich echt Spaß machen kann.
- Punkteabzug für Fehlende Fehlerbehandlung scheint ist manchmal ein bisschen viel. Ziemlich hoher Zeitaufwand für die Übung, besonders da die Schnittmenge zwischen Vorlesung und Übung nicht besonders groß ist. Würde gerne mehr theoretische Aufgaben sehen.
- T23: Übung ist sehr trocken und abtunnend
- Teilweise wird leider Vorwissen aus Pfp benötigt, dass andere Studiengänge vorher nicht hatten.
Teilweise hatten einige Tutoren sich nicht gut auf die neue Aufgabe der ‚mach‘ vorbereitet.
- Teilweise zu starke Abweichungen zur Vorlesung.
- Tutor Stefan Schöninger: fünf Minuten vor Übungsbeginn da zu sein, wäre schön. Auch um die Übung nicht gleich gehetzt zu beginnen
- Umstellung auf GIT statt SVN
- Vielleicht kann man kennzeichnen, welche Rechnerübungen eher für SPII und welche eher für SPI geeignet sind. Einige Tutoren hatten von den Aufgabenstellungen in SPI keine Ahnung. Sie hatten sich diese nicht angeschaut, weil sie ja SPII Tutoren waren.
- Wenn man keine Erfahrung in C hat kommt einem die Sprache sehr unübersichtlich vor. Bei manchen Aufgaben hätte ich mir eindeutiger Hilfestellungen in den Übungsfolien gewünscht.
- generell: bitte erklärt einmal ordentlich wie man wann fflush() verwendet! Das hat sich bei jeder Abgabe geändert ob es erwartet wird oder nicht
- weniger Strenge Hausaufgaben

4.3) Zur Lehrveranstaltung möchte ich im Übrigen anmerken:

- -Schade, dass es für die Übungsaufgaben keine ECTS gibt. Der Aufwand ist (insbesondere für Nichtinformatiker, die kein GTI, GRA und PFP hatten) sehr hoch
-Manchmal werden Punkte abgezogen, obwohl man als Nichtinformatiker nicht wissen kann, dass an dieser Stelle etwas kaputt gehen kann. Von "Underflow", der wohl in GTI behandelt wird, habe ich beispielsweise bis mir dafür Punkte abgezogen wurden noch nie gehört
- Bezieht sich auf die Übung Mittwoch 12:00-14:00
- D ist halt die Übung mit den Aufgaben, und die Theorie in der Vorlesung. Für die Klausur brauchen wir beides, aber Vorlesung und Übung gehen hier oft nicht in hand in hand mit einander
- Der Tutor macht seinen Job sehr gut, jedoch ist der Aufbau der Übung in SP in meinen Augen einfach nicht gut.
Die Übung sollte dafür da sein, Aspekte aus der Vorlesung noch mal verständlich zumachen und diese praktisch zu verinnerlichen. Im Moment wirkt die Übung wie eine völlig andere Veranstaltung als die Vorlesung.
- Eine zeitnahe Korrektur der Abgaben wäre wünschenswert.
- Ich habe jede Woche die Übung besucht und mir immer viel Zeit für die Hausaufgaben genommen. Im Schnitt habe ich ca. 10 bis 12 Stunden für die größeren Programmieraufgaben aufgewendet. Ich finde den Aufwand, um in den Hausaufgaben gut abzuschneiden schon sehr groß. Wenn man dabei bedenkt, auf wie viele verschiedene Arten man Fehler begehen kann. Vergessenes perror(), flag im Makefile vergessen etc. Das Programm ist dadurch meistens nicht gefährdet und ich denke, ein Kommentar würde mir da reichen

um den Fehler zum Beispiel in der Klausur nicht mehr zu wiederholen.

Ich kann mir aber vorstellen, dass unter Umständen den Leuten die Lust aufs Programmieren vergeht, wenn die erreichten Punkte im Schnitt nicht dem Aufwand entsprechen. Damit will ich nicht sagen, dass viel Aufwand == viele Punkte sein soll.

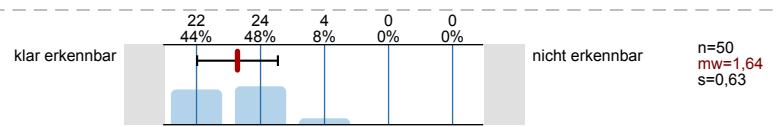
- Johannes Zahn: Leider teilweise schlechter vorbereitet und Korrekturen erst immer kurz vor der neuen Übung, so dass es nicht mehr möglich war sich diese vor der Besprechungsübung anzuschauen.
- Kilian ist cool :)
- Simon: war teilweise etwas schnippisch...solche Kommentare wie 'wenn man's nicht kann, sollte man's halt lassen' sind echt unnötig, schließlich sind wir da um was zu lernen.
Es hören nicht nur Informatiker SP, weswegen es nicht selbstverständlich ist dass man GTI, GRA und Co gehört hat und perfekt mit Linuxsystemen arbeiten kann
- Sollte viel konkreter mit der Vorlesung abgestimmt werden. Vorlesung ist zu 85% ohne Bezug zur Übung und kann man sich eig. sparen.
Eva Dengler könnte mehr auf Emails reagieren, meistens bekommt man eine verzögerte (3 Tage und mehr) oder keine Antwort.
- T23: Der Tutor wirkt engstjert aber nicht motivierend
- Tutor Donnerstag 10-12 Uhr (Julian?): Meist sehr unfreundlich, konnte keine sachliche Kritik am Code üben und nicht einfach nur helfen.
- Übung Ruben: Sehr kompetente Vermittlung des Stoffes
- Übungen wurden sehr gut gehalten. Johannes Weidner ist ein super Tutor!
Hammer Erklärungen!

5. Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter

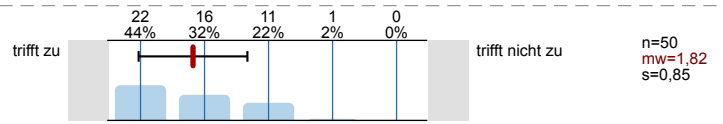
5.1) Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter beantworten?



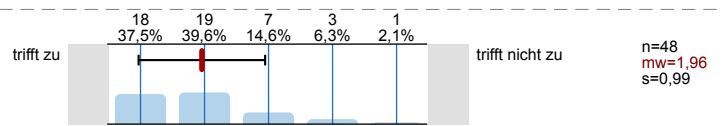
5.2) Zielsetzungen und Schwerpunkte des Übungsinhalts sind:



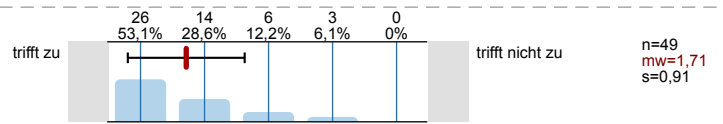
5.3) Ich werde gut zum selbstständigen Lösen von Aufgaben angeleitet.



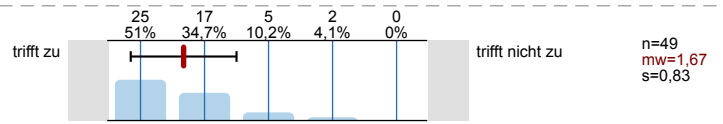
5.4) Die Anwendbarkeit des Übungsstoffes wird z.B. durch Beispiele gut verdeutlicht.



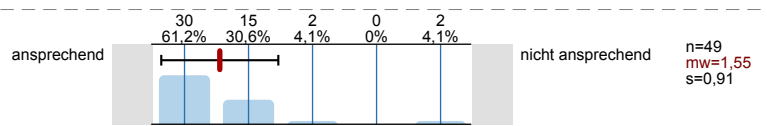
5.5) Die Übungsform (Aufgabenbehandlung, Programmieren, etc.) ist gut zur Vermittlung des Stoffes geeignet.



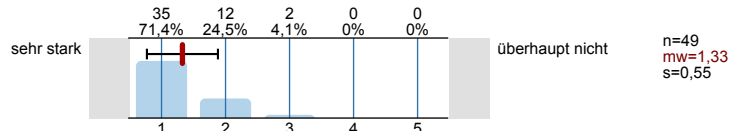
5.6) Die Präsentation von Aufgaben und Lösungen ist nachvollziehbar, es ist genügend Zeit zum Mitdenken vorhanden.



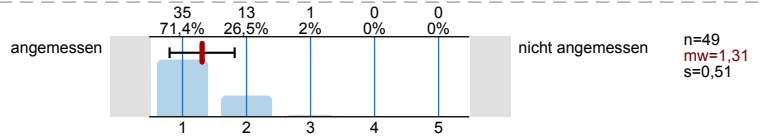
5.7) Der Präsentationsstil der Übungsleiterin/des Übungsleiters ist:



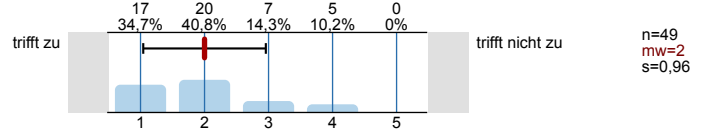
5.8) Die Übungsleiterin/Der Übungsleiter geht auf Fragen und Belange der Studierenden ein.



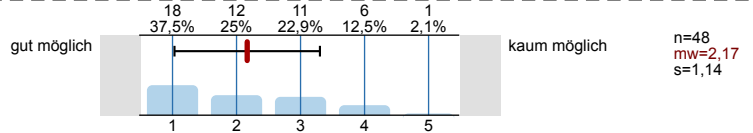
5.9) Der Einsatz und das Zusammenspiel von Medien (Tafel, Overhead-Projektor, Beamer, etc.) ist:



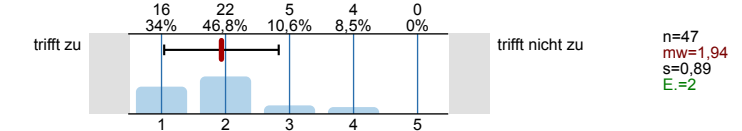
5.10) Die zur Verfügung gestellten Unterlagen sind in Menge und Qualität den Zielen der Übung angemessen.



5.11) Anhand des erarbeiteten Übungsmaterials ist die Vertiefung des Vorlesungs-/Modulinhalts:

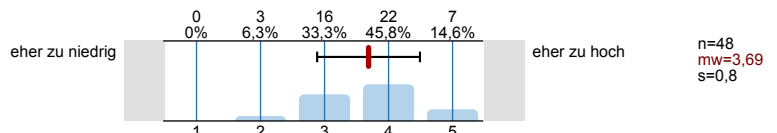


5.12) Der Bezug zu den Prüfungsanforderungen wird hergestellt.

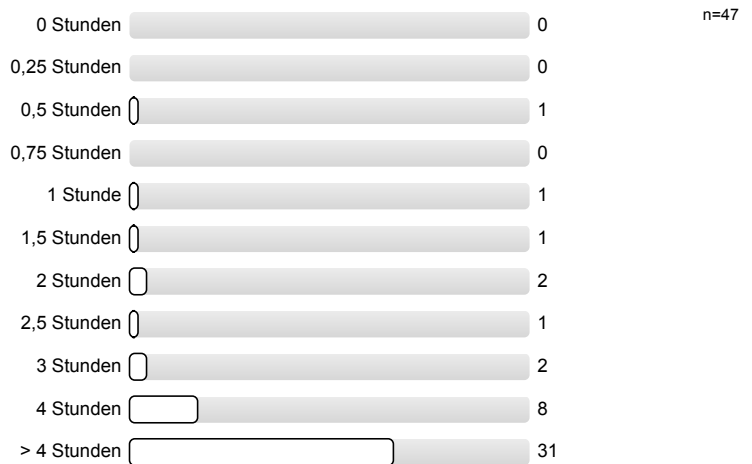


6. Schwierigkeitsgrad und Aufwand

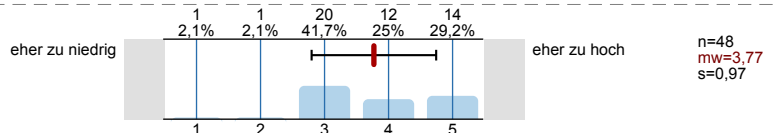
6.1) Der Schwierigkeitsgrad der Übung ist:



6.2) Mein Durchschnittsaufwand für Vor- und Nachbereitung dieser Übung beträgt pro Woche:

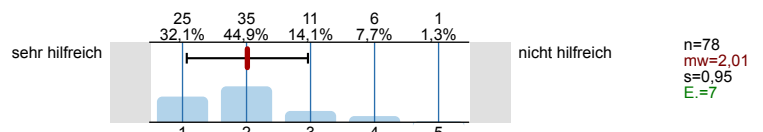


6.3) Meinen zeitlichen Durchschnittsaufwand für diese Übung finde ich:

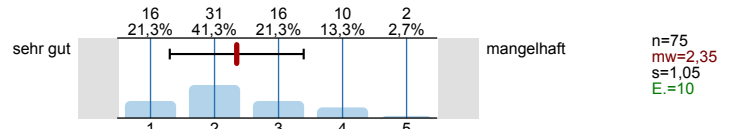


7. Von der Übungsleiterin/Vom Übungsleiter gestellte Fragen

7.1) Wie hilfreich fanden Sie die Korrekturhinweise auf der Webseite?



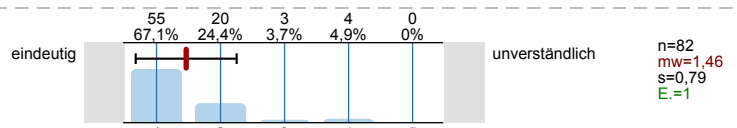
7.2) Wie gut haben Sie die Korrekturhinweise auf der Webseite über die Fehlerbehandlung bei Ein-/Ausgabe informiert?



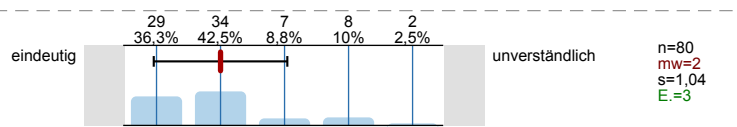
7.3) Anmerkungen zu den Korrekturhinweisen auf der Webseite

- An sich gut, werden von den Tutoren aber leider nicht immer berücksichtigt
- Braucht man ein free wenn realloc fehlschlägt?
- Der zweite Anmeldeschritt, der nötig ist, um an die pdf-Datei mit der Korrektur zu kommen hat sehr oft zu Problemen geführt. (Besonders, da das Kennwort zum svn-Verzeichnis ein anderes ist, als das, das ich auf der Website verwendet habe.) Wenn man schon im Waffel angemeldet ist, sollte man nicht noch einmal ein weiteres Passwort eingeben müssen, um die pdf-Datei mit der Korrektur herunterladen zu können.
- Die Bewertung nach Fehlern anstatt nach korrektem Code führt zu eindeutig unverhältnismäßigem Punktabzug. Wenn eine Aufgabe beispielsweise 300 Zeilen vollkommen korrekt funktionierenden Code enthält und 3 Zeilen Fehlerbehandlung fehlen, ist ein Abzug von bis zu 1,5 Punkten - also etwa 10% - viel zu viel für 1% falschen Code. Ich würde vorschlagen entweder für funktionierende Codeabschnitte Punkte zu vergeben, oder aber zumindest den Punktabzug in ein Verhältnis mit dem Umfang der Aufgabe zu setzen.
- Es wäre praktisch wenn die Links zur Korrektur in Waffel funktionieren würden
- Fehlte Information darüber, dass primär POSIX Pages gelten (vielleicht in Materialien, ... statt printf(3) printf(3p) schreiben würde Anforderungen klarer machen
- Korrekturhinweise sind sehr sinnvoll, um Transparenz bei der Bewertung zu garantieren.
- Mir war bei keiner Aufgabe so richtig klar wo Freiraum beginnt und Referenz endet. Gestalte ich sämtliche Ausgaben wie die Referenz? Setze ich die maßgeblichen Punkte um, muss mich dann aber wundern, dass mir Punkte abgezogen werden für Sachen die ich nicht für wichtig erachtet habe? Dabei helfen die Korrekturhinweise leider nicht.
- Obwohl ich sie nicht verwendet habe, finde ich diese überaus transparente Art der Korrektur super!
- Passt eigentlich so, man kann ja auch nicht den Quelltext direkt hinschreiben, sonst lernt keiner etwas.
- Viel zu penibel

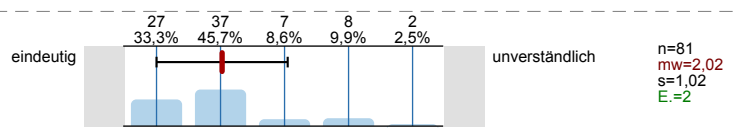
7.4) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 1: lilo?



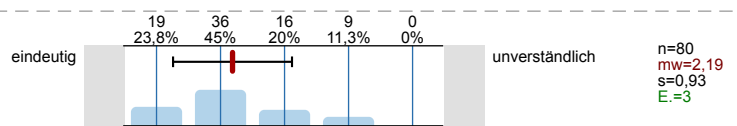
7.5) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 2: wsort?



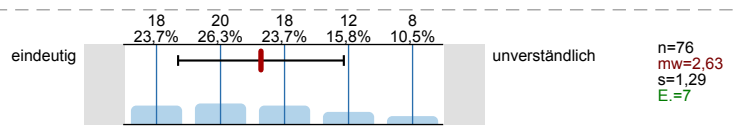
7.6) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 3: halde?



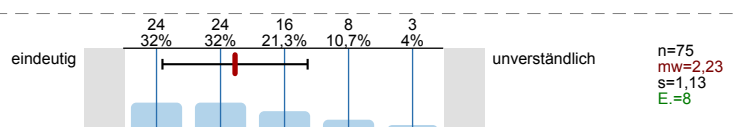
7.7) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 4: clash?



7.8) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 5: mach?



7.9) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 6: creeper?



7.10) Weitere Anmerkungen zu den Aufgaben

- "machfile" evtl. anderen Inhalt geben, da dies teilweise etwas genervt hat beim Programmieren..

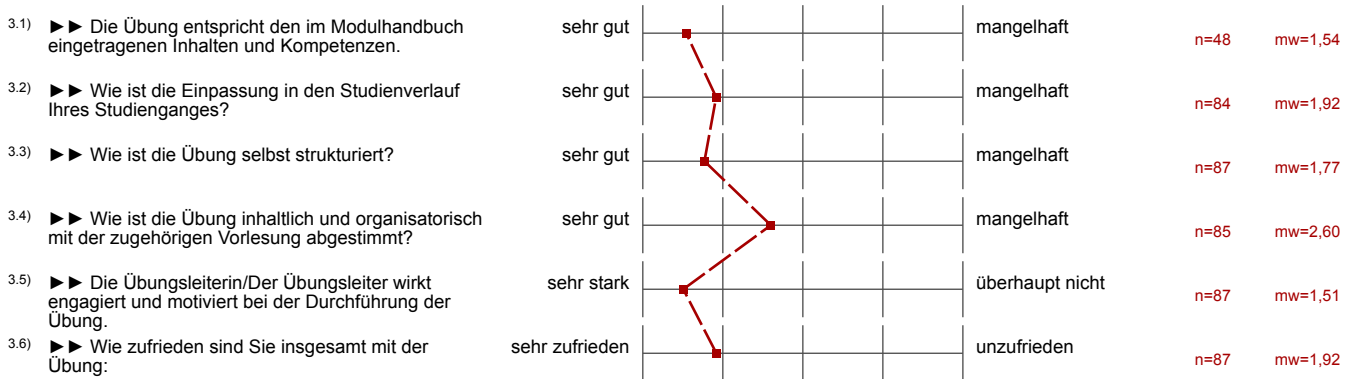
- Also gerade bei der Clash hatte ich viele "Philosophische" differenzen mit der Aufgabenstellungen. Als kleines Beispiel die Max Path Length. Ich gehe bei SP davon aus für ein Zielsystem zu entwickeln das da heißt debian. In debian sind das wohl 4096 Bytes. In windows sogar nur 260. Gut nun kann man mit Mounthen weiterer platten das ganze noch viel weiter treiben und den pfad länger machen, aber davon wurde weder in der Tafelübung noch in der Vorlesung etwas gesagt (Zumindest meines Wissens nach).
4 Punkte abzug dafür sind schon hart... Aber grade gegenüber den Menschen die nebenher arbeiten und keine Zeit haben in die Rechnerübung zu gehen um sich die richtige Lösung zu erbetteln ist die Korrektur einfach unfair
- Am Anfang fühlt man sich ohne Vorkenntnisse in c und im Umgang mit der bash schon sehr ins kalte Wasser geschmissen, deswegen geben viele direkt wieder auf. Aber dafür ist der Aha-Effekt nach ein paar Wochen umso größer:D
- Bewertung zu streng, 3 Punkte Abzug bei einer 12 Punkte Aufgabe wegen einer einzigen vergessenen Fehlerbehandlung
- Der Start in das Fach könnte besser unterstützt werden
- Die Aufgaben sind durchaus anspruchsvoll. Leider ist es sehr demotivierend, wenn ein funktionierendes Programm mit vielen Stunden Arbeit sehr schlecht bewertet wird, weil Fehlerbehandlung fehlt. Diese ist praktisch gesehen oft unnötig und verkopft.
- Die Aufgabenbeschreibung ist teilweise sehr kurz gehalten. Hier würde 1-2 Hinweise mehr helfen um zu verstehen, was von einem verlangt wird. Insbesondere wenn noch nicht man die Tutoren die Aufgaben verstanden haben.
- Die Rückgabe der Korrekturen ist unnötig umständlich (Das PDF als Link wäre ein guter Anfang)
- Die vorgegebenen lösungen haben teils sehr mangelhafte Fehlerbehandlung (z.B. fflush bei creeper ... nicht auszudenken was passiert wenn man seine 1000gb Festplatte voll hat und dann seine Suche nach einer Datei in eine Datei umlenken will, da merkt man dann nicht mal dass etwas schief gegangen ist!!)
- Ein paar mehr bzw. ausführlichere Beispiele würden bei der Bearbeitung sicherlich helfen
- Ich denke es ist nicht sehr sinnvoll mit dem Programm, das gerade ein File ausführen soll, sich selber nochmal zu bauen(-> mach mit machfile)
- Man kann sehr schnell, sehr viele, Punkte verlieren, indem man Fehlerbehandlung vergisst. Es ist fast nur mithilfe der Rechnerübung möglich eine höhere Anzahl an Punkten zu bekommen.
- Scheinen mehr dazu zu dienen C zu lernen als theorie zu vertiefen, in vielen Aufgaben ist die anwendung des eigentlichen Stoffen ziehmlich trivial
- Siehe Anmerkungen oben. Versteht mich nicht falsch. Ich möchte keine Tests à la AuD - Bitte Bitte nicht - aber die Freiheit in der Aufgabenstellung bringt ihre Tücken mit sich. Was wichtig ist und nicht ist nicht immer klar ersichtlich.
- Vor allem die Aufgabenstellungen von den letzten drei Blättern war etwas unverständlich und es kommt nicht wirklich immer gut raus aus was er ankommt.
- Zu 5:mach man hätte die Semaphore in der Übung noch ein wenig länger besprechen können
- abweichende referenzen
- sehr starker Fokus auf IO - vielleicht etwas mehr Algorithmik statt immer wieder Strings einlesen einführen?

Profillinie

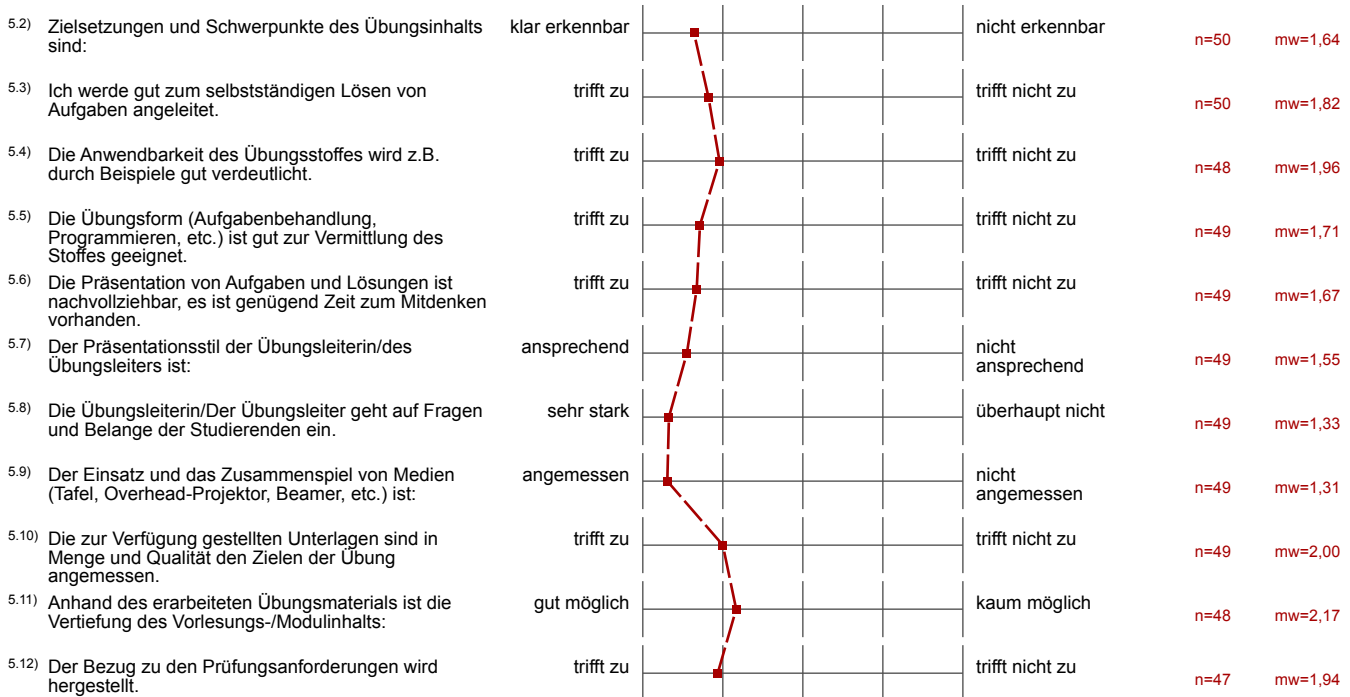
Teilbereich: Technische Fakultät (TF)
 Name der/des Lehrenden: M. Sc. Simon Ruderich
 Titel der Lehrveranstaltung: Übungen zu Systemprogrammierung 1
 (Name der Umfrage)

Verwendete Werte in der Profillinie: Mittelwert

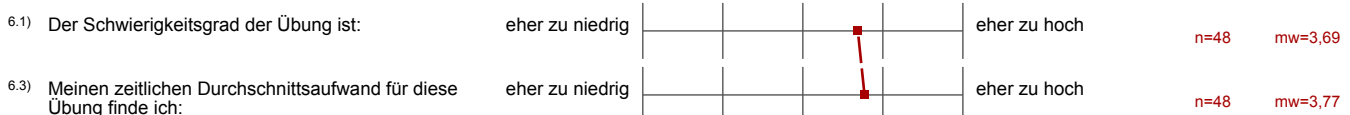
3. Hauptfragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter











5. Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter



6. Schwierigkeitsgrad und Aufwand



7. Von der Übungsleiterin/Vom Übungsleiter gestellte Fragen

7.1) Wie hilfreich fanden Sie die Korrekturhinweise auf der Webseite?	sehr hilfreich		nicht hilfreich	n=78	mw=2,01
7.2) Wie gut haben Sie die Korrekturhinweise auf der Webseite über die Fehlerbehandlung bei Ein-/Ausgabe informiert?	sehr gut		mangelhaft	n=75	mw=2,35
7.4) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 1: lilo?	eindeutig		unverständlich	n=82	mw=1,46
7.5) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 2: wsort?	eindeutig		unverständlich	n=80	mw=2,00
7.6) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 3: halde?	eindeutig		unverständlich	n=81	mw=2,02
7.7) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 4: clash?	eindeutig		unverständlich	n=80	mw=2,19
7.8) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 5: mach?	eindeutig		unverständlich	n=76	mw=2,63
7.9) Wie klar fanden Sie die Aufgabenstellung für Aufgabe 6: creeper?	eindeutig		unverständlich	n=75	mw=2,23