



Herr  
M. Sc. Christian Eichler (PERSÖNLICH)

## SS'17: Auswertung für Übungen zu Systemprogrammierung 1

Sehr geehrter Herr M. Sc. Eichler,

Sie erhalten hier die Ergebnisse der automatisierten Auswertung der Lehrveranstaltungsevaluation im SS'17 zu Ihrer Umfrage vom Typ "Übung":

- Übungen zu Systemprogrammierung 1 -

Es wurde hierbei der Fragebogen - t\_s17u58 - verwendet, es wurden 61 Fragebögen von Studierenden ausgefüllt.

Die Note 1 kennzeichnet hierbei eine maximale Güte, die Note 5 eine minimale Güte für die einzelnen Fragen bzw. Mittelwerte.

Der Kapitel-Indikator für "Hauptfragen zu Lehrveranstaltung und Dozentin/Dozent" zeigt den Mittelwert der 6 Hauptfragen und damit den Lehrqualitätsindex (LQI), dieser wird für die Bestenlisten der verschiedenen Kategorien, und zur Qualitätssicherung durch die Studienkommissionen verwendet.

Der Kapitel-Indikator für "Weitere Fragen zu Lehrveranstaltung und Dozentin/Dozent" 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. 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'17 möglich, siehe Bestenlisten, Percentile, etc.

Bitte melden Sie an [tf-evaluation@fau.de](mailto: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](mailto:kai.willner@fau.de))

Jürgen Fricke (Evaluationskoordinator, [tf-evaluation@fau.de](mailto:tf-evaluation@fau.de))

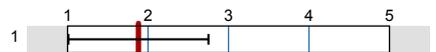
M. Sc. Christian Eichler

SS'17 • Übungen zu Systemprogrammierung 1  
 ID = 17s-Ü SP1  
 Rückläufer = 61 • Formular t\_s17u58 • LV-Typ "Übung"



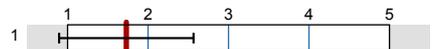
Globalwerte

3. Hauptfragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter



mw=1,88  
s=0,87

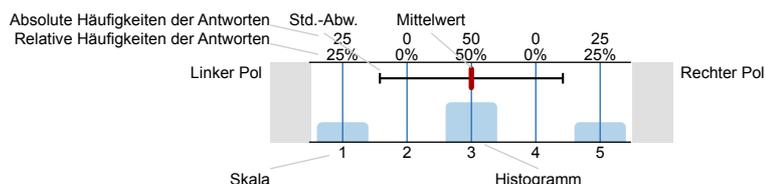
5. Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter



mw=1,73  
s=0,84

Legende

Fragetext



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

2. Allgemeines zur Person und zur Lehrveranstaltung

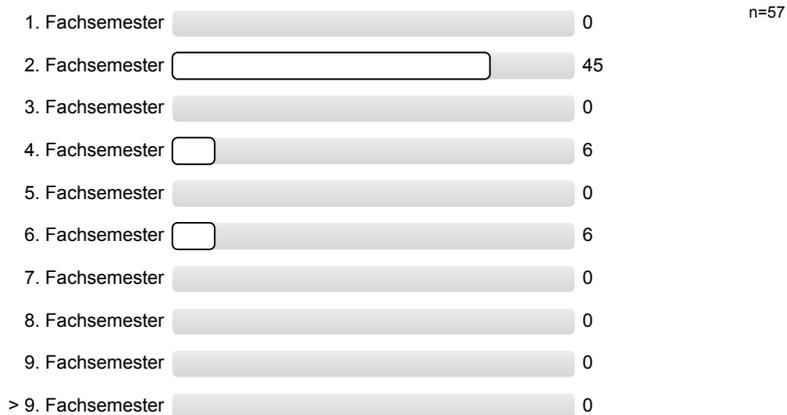
2.1) ▶▶ Ich studiere folgenden Studiengang:

CE • Computational Engineering	<input type="checkbox"/>	3	n=61
EEL • Elektrotechnik - Elektronik - Informationstechnik	<input type="checkbox"/>	3	
INF • Informatik	<input type="checkbox"/>	38	
INFLA • Informatik für Lehramt	<input type="checkbox"/>	1	
IuK • Informations- und Kommunikationstechnik	<input type="checkbox"/>	8	
MT • Medizintechnik	<input type="checkbox"/>	2	
WINF • Wirtschaftsinformatik	<input type="checkbox"/>	5	
Sonstiges	<input type="checkbox"/>	1	

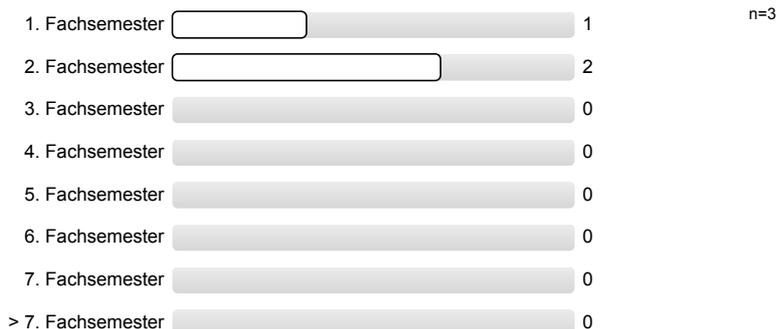
2.2) ▶▶ Ich mache folgenden Abschluss:

B.Sc. • Bachelor of Science	<input type="checkbox"/>	56	n=61
M.Sc. • Master of Science	<input type="checkbox"/>	3	
M.Sc.(hons) • Master of Science with Honours	<input type="checkbox"/>	0	
M.Ed. • Master of Education	<input type="checkbox"/>	0	
LA • Lehramt mit Staatsexamen	<input type="checkbox"/>	1	
Dr.-Ing. • Promotion	<input type="checkbox"/>	0	
Zwei-Fach-Bachelor of Arts	<input type="checkbox"/>	1	
Sonstiges	<input type="checkbox"/>	0	

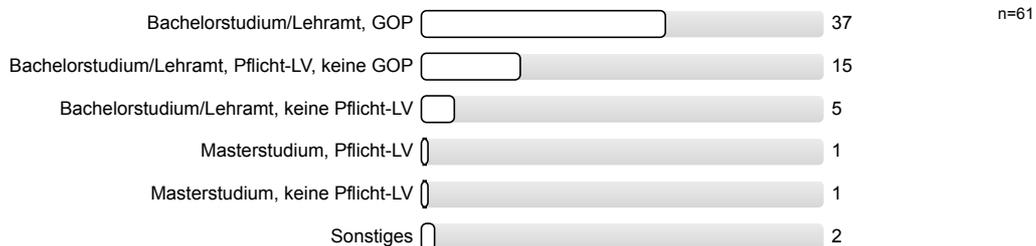
2.3) Ich bin im folgenden Fachsemester (im Bachelor):



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



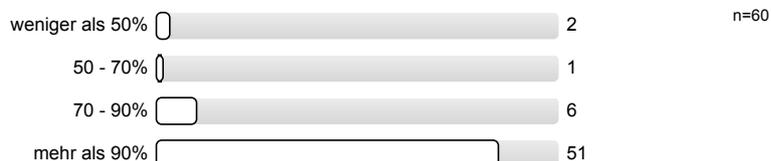
2.5) ►► Diese Lehrveranstaltung gehört für mich zum . . . .



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

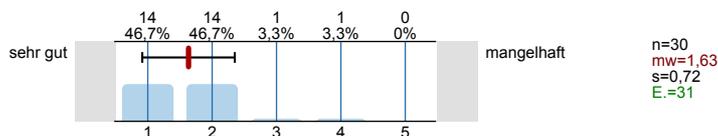
■ Mathematik

2.7) Ich besuche etwa . . . . Prozent dieser Übung.

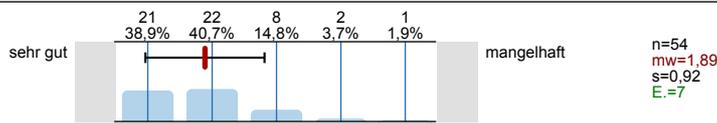


### 3. Hauptfragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter

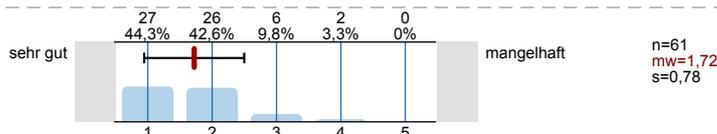
3.1) ►► Die Übung entspricht den im Modulhandbuch eingetragenen Inhalten und Kompetenzen.



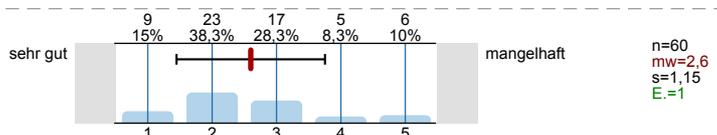
3.2) ►► Wie ist die Einpassung in den Studienverlauf Ihres Studienganges?



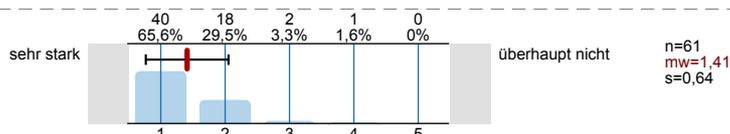
3.3) ►► Wie ist die Übung selbst strukturiert?



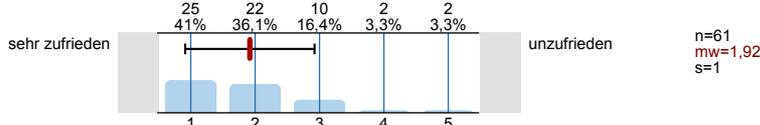
3.4) ►► Wie ist die Übung inhaltlich und organisatorisch mit der zugehörigen Vorlesung abgestimmt?



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:

- (T19) Man bekommt teilweise nützliche Tipps jenseits der Übungsfolien.  
 Programmierbeispiele aus den Übungen werden per E-Mail versandt.
- - Stringverarbeitung kann man wenigstens danach.  
 - Man lernt, misstrauisch zu werden, wenn man mal keinen Spezialfall hat.

@Lukas Lehnert{

- Man merkt, dass es schon gut auskennt mit C.
- Man schnell hilfreiche Antworten bei Fragen, die man per E-Mail stellt.

}

■ Anna Feiler:

Ich finde es zwar an sich gut, dass du laut und deutlich redest, aber etwas leiser wäre meiner Meinung nach besser. :D

■ Chris Eichler: Ein toller Übungsleiter. Zieht den Stoff bzw. die Besprechung der Aufgabe nicht unnötig in die Länge, so wie Prof. Kleinöder. Es ist schön, mal früher aufzuhören.

Prof. Kleinöder: Beim Arbeiten mit Code, sei es Beispiele, oder Codevorstellungen der Studenten in der Übung, bitte syntax highlighting verwenden! Der weiße Hintergrund und die schwarze Schrift sind wirklich eine Qual! Da lobe ich mir den Chris Eichler mit syntax highlighting im Terminal und in Vim + diverser Plugins in Vim, welche das gemeinsame Programmieren vereinfachen. Eine Man page farbig zu lesen macht einfach mehr Spaß!

■ Chris/Juk: viel gemeinsames Programmieren, v.a. dass clash nochmal gemeinsam gemacht wurde

■ Christian Strate: Erklärt gut wo genau Stolpersteine liegen, wo man aufpassen muss und erklärt das ganze an (nahezu) perfekt gelungenen Tafelbildern.

Lukas Lehnert: Rechnerübungen sind top!

Anna Feiler: In den Rechnerübungen wirkt sie meist ein wenig überrascht von den Aufgaben, hilft aber immer weiter!

■ Christian Strate: Kompetenter Tutor, der sich gut mit C auskennt.

■ Der Tutor kann den vorgestellten Stoff wirklich gut.

--Tutor: Dominik Paulus

■ Die guten Einblicke in die Systemprogrammierung

- Dominik Paulus
  - Er geht auch auf Dinge ein, die nicht auf den Folien stehen, vor allem was mögliche Fehlerquellen betrifft. (z.B. rechnen mit Pointern)
  - Man merkt, dass er ziemlich viel Ahnung von C hat.
- Dominik Prossel: sehr freundlicher und sympathischer Übungsleiter. Er ist fachlich gut, aber nicht zu detailverliebt wie manch anderer Übungsleiter. Bitte auch für SP2 verpflichten ;)
  - Dominik erklärt sehr gut und gibt hilfreiche Tipps.
  - Eichler ist echt top notch. Macht den Stoff schnell und verständlich durch. Das Programmieren am Ende von der Übung finde ich sehr gut. Da wird auf viel gemachte Fehler hingewiesen etc.
    - Eichler ist auch echt nett und hilft gerne weiter. Und ich finde ihn witzig ^^
  - Florian Güthlein geht auf jede Frage genau ein und beantwortet auch Fragen, die über den eigentlichen Inhalt hinaus gehen.
  - Geimeinsame Code-Erarbeitung hilft über Unsicherheiten hinwegzukommen.
    - Es wird auf alle Fragen eingegangen. Es wird auch mal was außenrum erklärt.
  - Gute Erklärungen
  - Helene Gsänger:
    - Sehr praktische Aufgaben, gute Erklärung der Aufgaben damit man diese eigenständig lösen kann
  - Informationen zu Vim. (Christian Strate)
  - Motivierte und gut verstaendliche Darstellung (Michael Panzlaff)
  - Optimismus
  - Preisi ist sexy
  - Preisi ist sexy.
  - RÜs: alle Tutoren sehr hilfsbereit und gut vorbereitet.
    - T11: Übungsleiterin hat sich Zeit genommen, auch mal Themen noch genauer zu erklären, die in den offiziellen Folien nicht vorkamen
  - Simon Bayer:
    - Übungen wurden immer in angemessenem Tempo gehalten
    - Auf Fragen wurde genügend eingegangen
  - T06: Mir gefällt, dass Gregor sich die Zeit nimmt auf Fragen ein zu gehen und diese zu beantworten.
  - T16: netter Tutor
  - T18: Lustige Tshirts des Tutors
  - Tafelübung 8: gemeinsames Programmieren auf der Leinwand
    - Tutor kennt sich aus und kann's erklären
    - Tutor macht sehr gute Übung, motiviert und bringt auch teilweise weiterführende Linux-Kenntnisse bei.
  - Zu den Rechneruebungen:
    - Sehr kompetente und engagierte Tutoren wie: Milan Stephan, Christian Schlumberger, Thomas Altstidl, Dominik Paulus und Hans-Peter Deifel
    - Zu der Tafeluebung bei Thomas Altstidl:
      - Interaktiv ein Beispiel programmieren und das Ergebnis per Mail zugesendet zu kriegen.
  - der Tutor hat Duplos an die Studenten verteilt, welche ihre Abgabe vorgestellt haben
    - interaktiv die Tafel bemalen hat viel zum Verständnis beigetragen
  - einfache Anwendungsbeispiele werden gleich in der Übung angesprochen und selbst programmiert

- Übungsleiter: Christian Strate

Der Tutor ist motiviert, erklärt gut und kennt sich aus

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

- - Es sollte mehr um grundlegendes Verständnis für die Sprache C gehen und nicht um unzählige Spezialfälle und Kleinigkeiten, die eher als Stolpersteine empfunden werden (z.B. leere Zeile ignorieren bei der words.c). Vielleicht wären mehrere kleine, weniger umfangreiche Aufgaben eine Idee.

@Lukas Lehnert{

- Wirkt oft etwas gehetzt und durcheinander, gerade was das Zeitmanagement angeht. Als Idee vielleicht vorher einen groben Zeitplan erstellen, wann welches Thema in der Übung behandelt wird.

- Erstmal bevor auf die ganzen Details der Aufgabe eingegangen wird, kurz die Aufgabe ganz allgemein beschreiben für 1-2 Minuten.

- Keine Codebeispiele oder Ähnliches an die Tafel schreiben, was explizit irrelevant für die Aufgabe ist. Auch wenn es nett gemeint ist, als "Blick über den Tellerrand", verwirrt es oft mehr als es hilft. Eventuell dies noch expliziter als "nicht relevant" markieren.

}

- Abstimmung Übung und Vorlesung (wie bereits sicher häufig erwähnt)

- Anna Feiler:

Du wirkst manchmal relativ unvorbereitet und gelangweilt.

Außerdem lässt du, wenn du Studenten bei Fragen aufrufst, sie teilweise nicht ganz ausreden, weil du merkst, dass sie das Richtige antworten. Das ist für andere Studenten, die es evtl. nicht verstehen, nicht so gut.

- Chris/Juk: basst ;)

- Christian Strate: Der Tutor war manchmal etwas übermüdet.

- Der Bezug zur Übung und Vorlesung ist meistens nicht sehr klar.

Wenn man vorher nicht Erfahrung in C hat, sind die Aufgaben nicht wirklich gut lösbar.

Der Aufwand für die Aufgaben ist relativ hoch.

- Der Zusammenhang zur Vorlesung ist halt (fast, und wenn nur sehr sehr spät) nicht gegeben.

Christian Strate: Der Tutor wirkt oft sehr übermüdet und erklärt wie er die Aufgaben die Nacht zuvor korrigiert hat, steigert vlt. nicht ganz die Motivation

- Die Übungsaufgaben nehmen viel zu viel Zeit in Anspruch und sind daher schlecht bis gar nicht mit dem Besuch von allen Lehrveranstaltungen des Studiengangs vereinbar.

- Gerade die ersten Übungen sind zu schwer, da es für die meisten die ersten Kontakte mit C sind, und die Aufgaben schon umfangreich und komplex

- Ich brauche Ewigkeiten für die Hausaufgaben und die svn-Abgabe ist die reinste Seuche vor allem mit Partner. Es wäre schön, wenn die Angaben zu den Hausaufgaben auch etwas weniger kryptisch wären.

- Könnte etwas interaktiver sein

- PREISI IST SEXY?!

- Preisi ist supersexy

- SVN könnte am Anfang etwas besser Erklärt werden

- SVN??? => GIT!!! (oder wenigstens EST)

EST auch bei der Anmeldung und nicht Waffel.

Die Aufgaben am Ende der Tafelübung (Gelerntes Anwenden) könnten gerne stärker behandelt werden.

- T06: Anfangs leicht nervös in der ersten Übung, wurde aber deutlich besser :)
- T19: Unsere Abgaben wurden seit der wsort nie rechtzeitig bis zum Besprechungstermin korrigiert, das macht die ganze Besprechung sinnlos und aufgrund der Anwesenheitspflicht zu verschwendeter Zeit. Ich habe im Allgemeinen auch den Eindruck, dass die Tutorin nicht sehr organisiert ist und gehe mal davon aus (tut mir Leid, falls das nicht stimmt), dass sie die Korrektur immer bis zu letzt aufschiebt, weshalb sie nie fertig wird.  
Außerdem waren die Korrekturhinweise teilweise ziemlich missverständlich ausgedrückt, wodurch man teilweise eine Weile braucht, bis man versteht, was man falsch gemacht hat.
- Weniger Abzüge für Folgefehler  
Mehr Punkte für die Aufgaben damit die Abzüge nicht so stark ins Gewicht fallen.
- Wenn die Übung noch ein bisschen stringenter gehalten und der rote Faden etwas mehr zum Vorschein kommen würde, wäre sie wohl perfekt...
- Wenn es blöd läuft, bekommt man kaum Punkte auf die Aufgaben, obwohl man praktisch sehr viel funktionierenden Code hat.  
Definitiv zu wenig Kuchen in dieser Übung. Wir brauchen mehr Kuchen.
- Ziegler: teilweise unrichtige/unsaubere Aussagen und Erklärungen; Präsentationsstil etwas wirr
- keine Rechnerübungen an Freitagen. Zu Stosszeiten zu wenige Tutoren
- svn – git verwenden!!!  
Dass der Inhalt von Fehlermeldungen (perror statt fprintf stderr oder umgekehrt macht ja nur einen Unterschied beim ausgegeben Inhalt) so starken Punktabzug bringen kann (also auch, wenn eigentlich alles funktioniert und die Fehler auch behandelt werden); stattdessen nur pauschal einen Punkt oder so abziehen, wenn es bei einer Aufgabe besonders oft vorkommt.
- zu wenig Interaktion mit Studenten möglich
- Übungsleiter: Christian Strate  
nichts, du machst das gut

4.3) Zur Lehrveranstaltung möchte ich im Übrigen anmerken:

- Allgemein: Insgesamt fand ich den Schwierigkeitsgrad viel zu hoch.
- Darf man eigentlich Arrays benutzen?
- Der Zeitaufwand pro Hausaufgabe ging locker über 8h hinaus. Wobei Rechnerübungen auch besucht worden sind.  
Wenigstens eine Rechnerübung mit Tutor am Freitag wäre wünschenswert.
- Der Übungsleiter wirkt zwar kompetent, ist jedoch zu wenig auf die entsprechende Übungsstunde vorbereitet.(Michael Panzlaff)
- Die Rechnerübungen haben sehr gut weitergeholfen, wenn man bei einer Aufgabe nicht weiterkam oder mit dem SVN Probleme hatte
- Die Tutoren der Rechner-Übungen geben sich viel Mühe, bestimmte Sachverhalte genauer zu erklären.
- Die stoffliche Trennung von Vorlesung und Übung ist nachvollziehbar, jedoch würde ich mir teilweise eine Wiederholung wichtiger Punkte aus der Vorlesung in den Übungen wünschen.
- Dominik Paulus kann gut anhand von Beispielen Probleme und Loesungsidee usw weitergeben
- Duplos erhöhen die Motivation
- Durch den ungünstig gewählten Termin der Übung hat sie der Vorlesung Inhalte vorausgegriffen und hat deswegen in meinen Augen ihr Ziel, bereits gelerntes zu festigen, verfehlt
- Es ist gut, dass man Übungen auch mal nachreichen kann.
- Es wäre hilfreicher sich mehr Zeit für den Einstieg in C zu lassen.
- Evtl. wäre eine Intensivübung (siehe AuD) zu Beginn angebracht in C, der Einstieg fiel mir persönlich sehr schwer darin und ging nur dank exzellenter und hilfsbereiter Komilitonen.
- Helene Gsänger:  
das die Übung recht spärlich besucht wurde lag am leider am Tag. Mittwochs gab es in diesem Semester keine Vorlesung (Informatik, 2.Semester), sodass viele sich denTag "freigenommen" haben. Schade denn die Übung war recht gut!

- Im Gegensatz zu anderen Fächern, bei denen man Punkte auf richtige Dinge bekommt, hier bekommt man Antipunkte für alles, was falsch ist

=> sehr viele Stunden Arbeit wird durch viele kleinere Fehler zu sehr wenig Punkten => könnte manch einen deprimieren

(ist jedoch gut in dem Sinne, dass man wirklich weiß, was alles wo genau wie falsch war!)

Einstieg in die ersten Übungen ist mir persönlich sehr schwer gefallen

- Julian Brandner

- Leider besteht durch die Inhaltliche Lücke zwischen Vorlesung und Übung fast kein Anreiz, sich den Vorlesungsstoff nochmal genauer anzuschauen.

Es wäre praktisch, wenn da noch z.B. Theoriefragen zur Nachbearbeitung vorhanden wären um den Stoff zu verinnerlichen.

Mir gefällt aber, wie die einzelnen Übungen aufeinander aufbauen und dass diese logisch wirklich durchdacht sind.

- Tafelübung:

Räume mit Projektoren auf die Tafel- und Wandseite, wie A 2.16, sind perfekt zum Vorstellen von Programmcode.

- Übungsleiter hat leider immer Aufgabenblätter vergessen

- Übungsleiter: Christian Strate

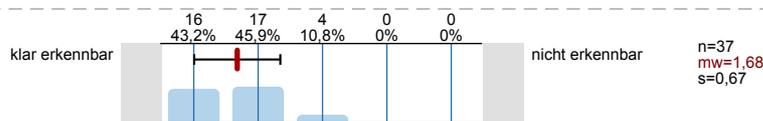
Die Übungsaufgaben und die Vorlesung haben manchmal kaum etwas miteinander zu tun (zumindest habe ich das Gefühl)

5. Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter

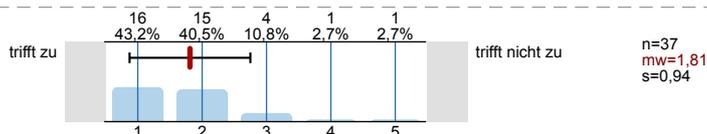
- 5.1) Weitere Fragen zu Lehrveranstaltung und Übungsleiterin/Übungsleiter beantworten?



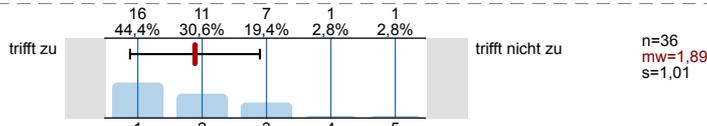
- 5.2) Zielsetzungen und Schwerpunkte des Übungsinhalts sind:



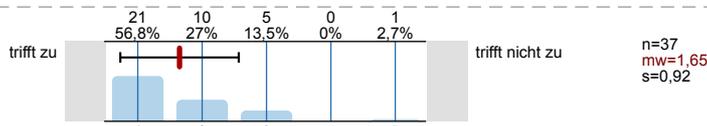
- 5.3) Ich werde gut zum selbstständigen Lösen von Aufgaben angeleitet.



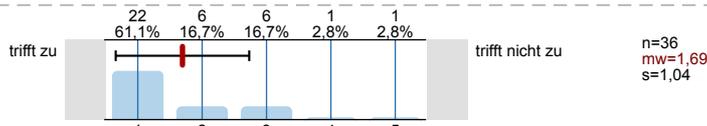
- 5.4) Die Anwendbarkeit des Übungsstoffes wird z.B. durch Beispiele gut verdeutlicht.



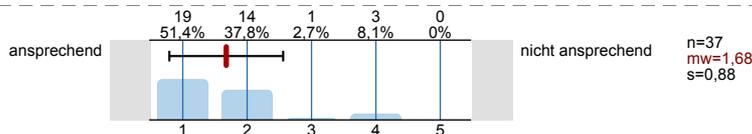
- 5.5) Die Übungsform (Aufgabenbehandlung, Programmieren, etc.) ist gut zur Vermittlung des Stoffes geeignet.



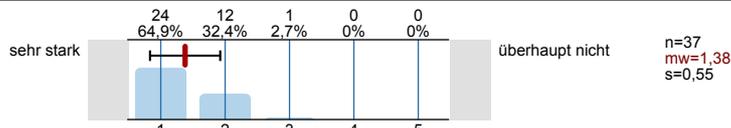
- 5.6) Die Präsentation von Aufgaben und Lösungen ist nachvollziehbar, es ist genügend Zeit zum Mitdenken vorhanden.



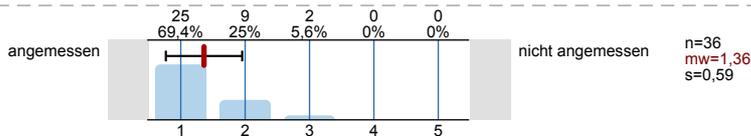
- 5.7) Der Präsentationsstil der Übungsleiterin/des Übungsleiters ist:



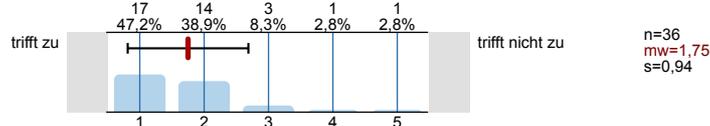
5.8) Die Übungsleiterin/Der Übungsleiter geht auf Fragen und Belange der Studierenden ein.



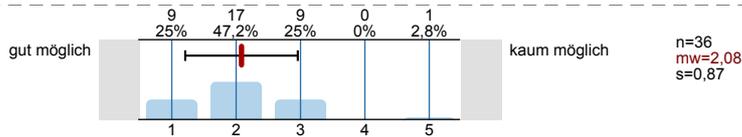
5.9) Der Einsatz und das Zusammenspiel von Medien (Tafel, Overhead-Projektor, Beamer, etc.) ist:



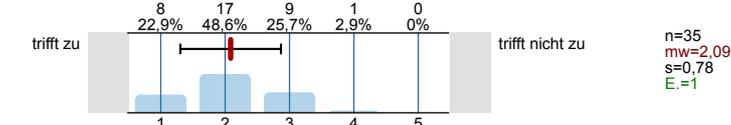
5.10) Die zur Verfügung gestellten Unterlagen sind in Menge und Qualität den Zielen der Übung angemessen.



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

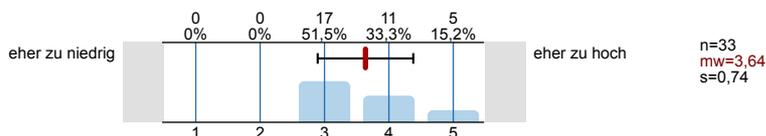


5.12) Der Bezug zu den Prüfungsanforderungen wird hergestellt.

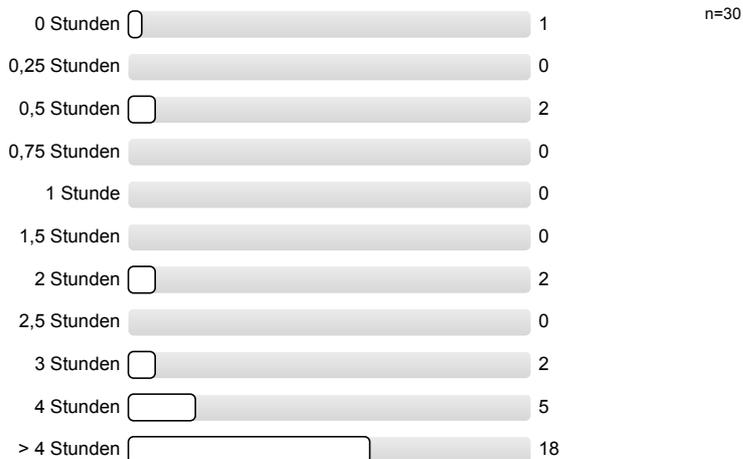


### 6. Schwierigkeitsgrad und Aufwand

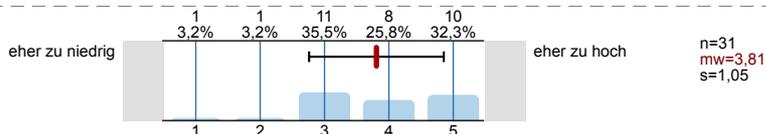
6.1) Der Schwierigkeitsgrad der Übung ist:



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

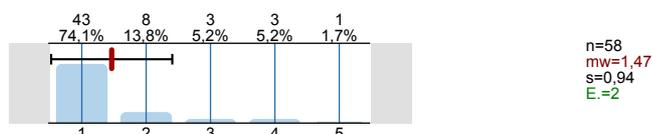


6.3) Meinen zeitlichen Durchschnittsaufwand für diese Übung finde ich:

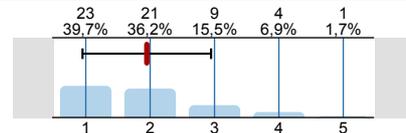


### 7. Von der Übungsleiterin/Vom Übungsleiter gestellte Fragen

7.1) lilo: Aufgabenstellung ist klar & eindeutig



7.2) wsort: Aufgabenstellung ist klar & eindeutig



n=58  
mw=1,95  
s=1  
E.=2

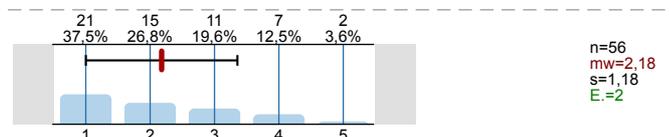
7.3) Wie empfanden Sie die Strenge der Korrektur?

- An sich ist die Korrektur sehr gut, ab und zu bekommt man aber doch weniger Punkte als erwartet, besonders wenn sich der ein oder andere Fehler eingeschlichen hat. Das ist ärgerlich, da man doch sehr viel Zeit in die Aufgaben investiert bzw. investieren muss!
- Angemessen
- Angemessen
- Angemessen
- Angemessen, obwohl wenn man 40% der Punkte durch wiederholte Fehler verliert es sehr ärgerlich ist, wenn man bedenkt, wie viel Zeit man in die Übung gesteckt hat
- Anna Feiler:  
Gut
- Basd scho. (Christian Strate)
- Bei all den eigentlich schier unmöglichen Sonderfällen, fuer deren Nicht-Pruefung es trotzdem Abzug gab, etwas zu streng
- Besonders bezüglich Inhalt der Fehler!-->ausgaben<--!! (nicht der Fehlerüberprüfung selbst) viel zu streng
- Bisschen zu streng, weil man bei Fehlern Punktabzug bekommt, aber nicht wirklich wieder Pluspunkte sammeln kann. Wenn man also ein funktionierendes Programm hat, dann kann es trotzdem sein, dass man so gut wie keine Punkte darauf bekommt, weil man evtl Fehlerabfragen vergessen hat, etc. (was natürlich trotzdem wichtig ist)
- Da es sich "nur" um Bonuspunkte handelt, in Ordnung.  
Lieber einen halben Punkt Abzug, als den Fehler nicht zu bemerken
- Dass es für Fehler Minuspunkte gibt, ist logisch, aber dass man auf ein Programm, das wie vorgeschrieben funktioniert, nur knapp über die Hälfte der Punkte bekommt, ist etwas überzogen .
- Die Korrektur ist gut und weißt gut auf alles hin und ist nahvollziehbar
- Eher streng! Besonders bei den ersten Übungen habe ich es so empfunden.  
Für jede Kleinigkeit gab es Punktabzug (bspws Fehlerausgaben in einem Bibliotheksmodul)
- Eher zu hart, da für einen Fehler oftmals mehre Punkte abgezogen wurden
- Es ist manchmal etwas nervtötend, wenn für jede Fehlerbehandlung gleich Punkte abgezogen werden.  
Es ist klar, dass durch die Sprachlichen Hürden von C diese notwendig sind, aber z.B. 1-1,5 Punkte pro Übung wären ausreichen. Schließlich geht es hier ja um das Konzept z.B. einer Kommandozeile o.ä. und nicht darum "Wie behebe ich Fehler am besten".
- Etwas zu hart
- Evtl teilweise etwas zu streng. Viele kleine Fehler können leider schon zu ner schlechten Prozentzahl führen obwohl der Code an sich "super" ist.
- Geil!
- Ich fand es gut, dass man auch Punkte bekommen hat, wenn z.B. bei der wsort das Einlesen nicht funktioniert hat, aber der Rest trotzdem nicht ganz so kaputt war. Alles in allem fand ich die Bewertung fair und nachvollziehbar.
- Ich finde die Korrektur allgemein schon sehr streng.  
Wenn das Programm zu 95% funktioniert (Außer in quasi nicht auftretenden Ausnahmen) erwarte ich auch eine vergleichbare Menge an Punkten. Bei wenigen Fehlern in diesen Ausnahmen können aber schnell die Hälfte der Punkte Weg sein, weil für jeden Fehler 1-2 Punkte abgezogen werden. Ich habe also eine gut funktionierendes Programm und nur 50% der Punkte. Das ist nicht sehr motivierend für die Aufgaben wenn man weiß, dass ich ein gutes Programm abgeben kann und trotzdem nur wenige Punkte bekomme. Vielleicht kann man in Zukunft die Fehlerpunkte davon abhängig machen wie sehr sie den Programmablauf stören.  
Programm tut nicht was es soll -> Viele Fehlerpunkte  
Programm läuft im allgemeinen gut -> Wenig Fehlerpunkte
- Im Allgemeinen in Ordnung, teilweise wurden jedoch Punkte für mangelnde Fehlerbehandlung abgezogen, wenn es nachweislich unmöglich ist, dass dieser Fehler an der Stelle auftritt, was ich bei aller Liebe zu robuster Programmierung nicht nachvollziehen kann.  
Was ich in dem Zusammenhang auch seltsam finde, ist, zu sagen, die Manpage im CIP sei für Fehlermeldungen Referenz und dann

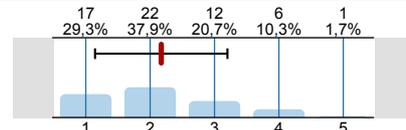
aber Punkte für ferror bei fgets abziehen und auf die Übungsfolien zu verweisen. Da soll dann meiner Meinung nach in der Übung aber auch explizit darauf verwiesen werden(was bei uns aber nicht der Fall war), dass in der Manpage nicht steht, dass die errno gesetzt wird, damit man sich nicht auf diese verlässt.

- Korrektur oft zu streng. Zu viel Punktabzug bei einfachen Flüchtigkeitsfehlern (Z.Bsp. "static" oder Fehlerbehandlung an einer Stelle vergessen, aber sonst vorhanden). Man ist froh, dass die eigenen Programme nach mehreren Stunden programmieren mit sämtlichen Debuggern fehlerfrei funktionieren und am Ende bekommt man doch nur etwas mehr als Hälfte der Punkte.
- Ok
- Okay
- Relativ streng
- Sehr streng
- Sehr streng, wenn es um Kleinigkeiten geht.
- Solange eine strenge Korrektur auch immer einem genau sagt, was warum falsch ist, ist daran an sich nichts auszusetzen solange die Aufgaben grundsätzlich freiwillig sind.
- T15: Korrektur fair und immer nachvollziehbar
- Teilweise unverständlich, so wurde zwar die Posix-Manpage angegeben, aber nach Linux-Manpage korrigiert (halde)  
 Alles in allem schon streng, das hat aber zu einem besseren Verständnis von C und den Fettnäpfchen der Sprache geführt, als das man jetzt zumindest dreimal alle Manpages durchschaut, ob es denn zu Fehlern kommen könnte.
- Teilweise wegen nicht ganz klarer Aufgabenstellung übertrieben
- War angemessen
- War in Ordnung
- War ok
- Was die Fehlerbehandlung angeht auesserst stark. Ansonsten fair.
- Zu streng
- Zu streng, für jeden kleinen Fehler, der keine direkten Auswirkungen auf die eigentliche Programmaufgabe hat, werden Punkte abgezogen
- Zu viel Punktabzug für kleine oder "bad practice" Fehler obwohl das Programm (oder die Makefile) funktioniert.
- akzeptabel - wenn auch penibel
- angebracht
- angemessen
- gut
- okay
- sehr streng
- sehr streng, aber muss halt sein damit mans richtig lernt
- streng
- unterschiedliche Tutoren haben unterschiedlich streng korrigiert. Teilweise war es sehr fair, bei der nächsten Aufgabe wurde dann sehr streng abgezogen
- zu viel Fokus auf auf fehlende Speicherfreigaben und Fehlerbehandlungen obwohl Program funktioniert.
- Übungsleiter: Christian Strate  
 Deine Korrektur ist sehr hart. Ich finde das aber gut, da man nur so etwas lernt. Auch wenn manch dich dafür nicht immer mögen, mach weiter so

7.4) halde: Aufgabenstellung ist klar & eindeutig

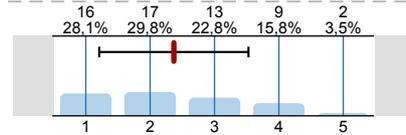


7.5) clash: Aufgabenstellung ist klar & eindeutig



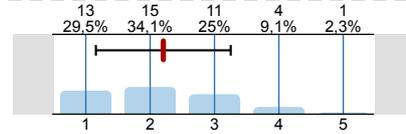
n=58  
mw=2,17  
s=1,03  
E.=2

7.6) piper: Aufgabenstellung ist klar & eindeutig



n=57  
mw=2,37  
s=1,16  
E.=2

7.7) crawl: Aufgabenstellung ist klar & eindeutig



n=44  
mw=2,2  
s=1,05  
E.=15

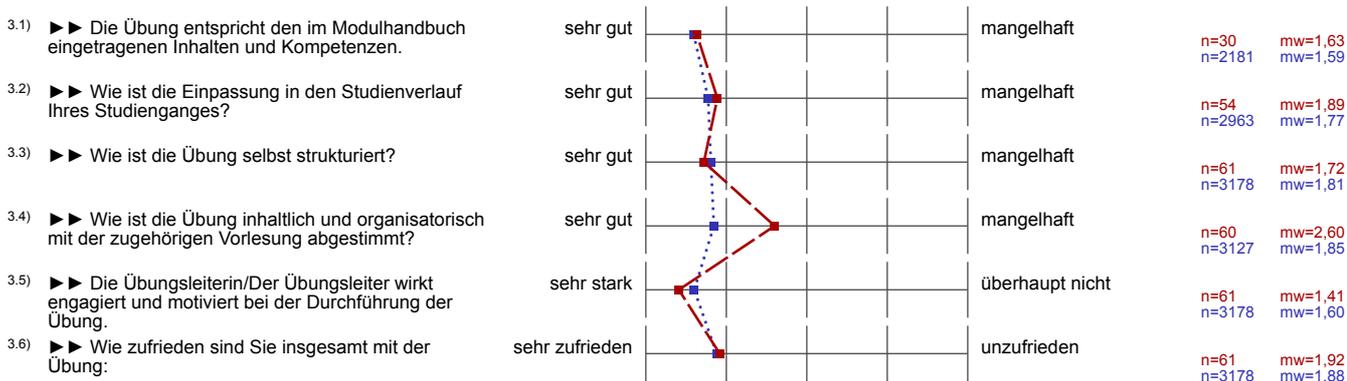
# Profillinie

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

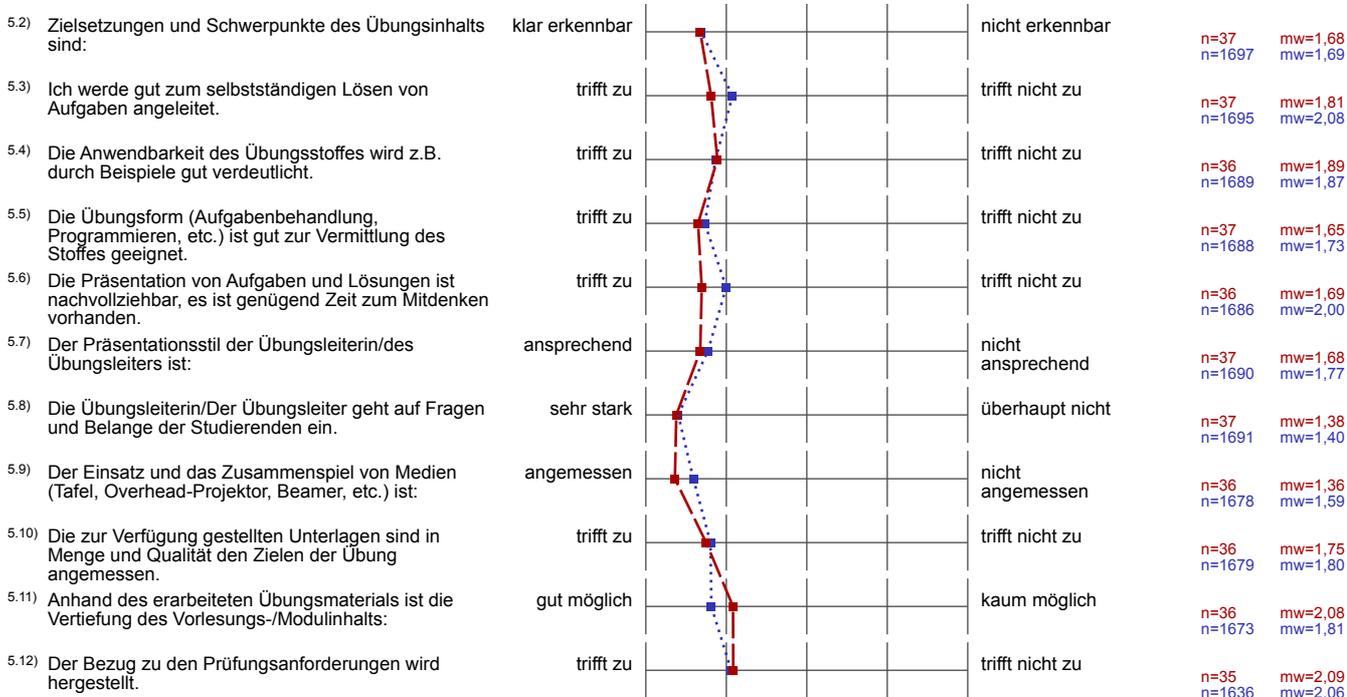
Vergleichslinie: Mittelwert\_aller\_Übungs\_Rückläufer\_SS17

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



