

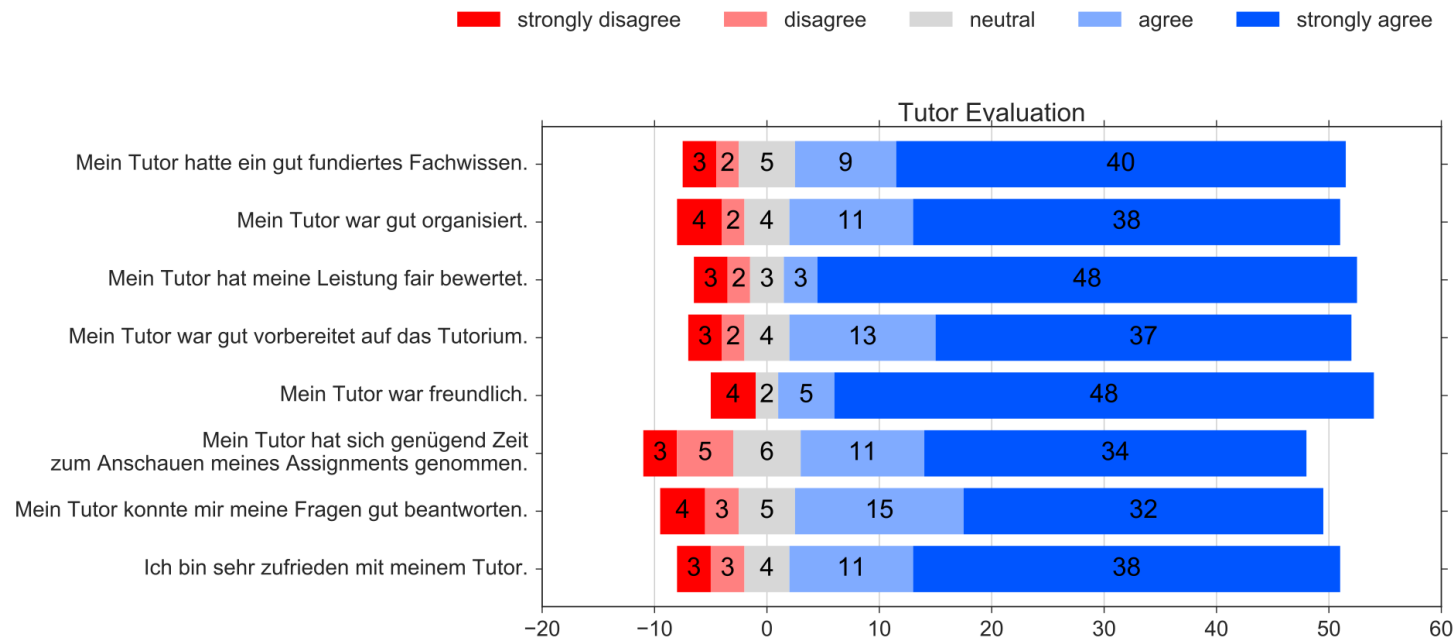
Programmieren 1

Auditorium Exercise 14

Evaluation der Tutorien 2018

- Ihr könnt eure Tutorien anonymisiert bewerten

Durchschnittsschulnote für die Tutoren: 1,73



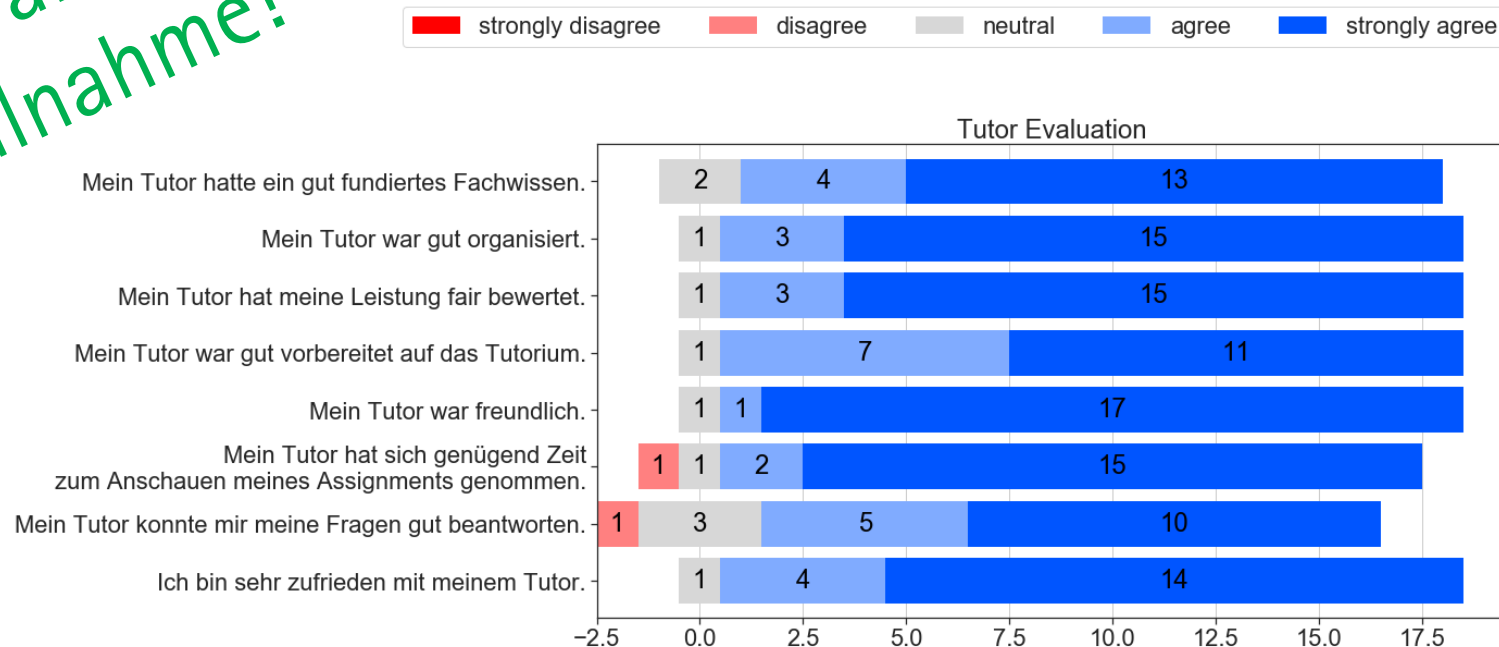
Daten vom 29.01.18 morgens, 59 Teilnehmer

Evaluation der Tutorien 2019

- Ihr könnt eure Tutorien anonymisiert bewerten

Vielen Dank für die
Teilnahme!

Durchschnittsschulnote für die Tutoren: 1,32



Daten vom 01.02 morgens, 19 Teilnehmer

Evaluation der Tutorien

- Ihr könnt eure Tutorien anonymisiert bewerten!
- Nehmt dafür bitte bis zum 04.02.19 an der Umfrage unter folgendem Link teil:

<https://iq4s.hci.uni-hannover.de/index.php?id=d59dcb4a>



Reversi Winterchallenge

Alle Teilnehmer der Winterchallenge:

- Mittwoch der 06.02.19 um 15:00 Uhr
- Appelstraße 9A, 9. Etage, Seminarraum 901

Ablauf:

- Turnier & Siegerehrung
- gemütliches Zusammensitzen mit ein paar Keksen und Getränken

STUDIENLEISTUNG & PRÜFUNG

Assignment Grading

- 0 points, if you don't do anything or no decent attempt at the assignment is made
- 1 point, if you properly solve at least one complete task (including all subtasks) from the assignment
- 2 points, if you complete all tasks in the assignment properly
- One point in every single assignment? -> you pass
- At least 21 points overall? -> in addition, you also get a bonus in the exam
 - Bonus applies only to the exams in WS 18/19 and SS19. No longer valid afterwards.

Studienleistung & Klausurbonus

Wann wird die Studienleistung eingetragen? Bonus?


- Plagiatsprüfung
- Eintragung der Studienleistung (~Mitte Februar)
- Daraufgehend laden wir in StudIP eine Liste hoch, in der steht ob ihr den Bonus erreicht habt
- Ihr habt die Möglichkeit im Assignmentssystem euch die Summe eurer vorläufigen Punkte anzusehen. (vgl. nächste Folie)

Bitte seht davon ab eure Tutoren nach den Punkten zu fragen!

Studienleistung & Klausurbonus

Ihr habt die Möglichkeit im Assignmentsystem euch die Summe eurer vorläufigen Punkte anzusehen.

Programmieren1 WiSe2017:

List of all results:			
Assignment	Upload Date	Status	Download
Assignment 1	-	✗	-
Assignment 2	-	✗	-
Assignment 3	-	✗	-
Assignment 4	-	✗	-
Assignment 5	-	✗	-
Assignment 6	-	✗	-
Assignment 7	-	✗	-
Assignment 8	-	✗	-
Assignment 9	18. December 2017 08:45:34	✓	
Assignment 10	-	✗	-
Assignment 11	-	✗	-
Assignment 12	-	✗	-
You achieved 0 points in total.			



Bitte seht davon ab eure Tutoren nach den Punkten zu fragen!

Klausur

- Termine: 03.04, 04.04 & 05.04
- Raum F411
- Zeit: 90min
- 4 Slots pro Tag je ~58 Personen (vorläufige Planung)
 - 03.04, 04.04 & 05.04: 8:00 – 10:00 Uhr
 - 03.04 & 04.04 & 05.04: 10:00 – 12:00 Uhr
 - 03.04 & 04.04: 13:00 – 15:00 Uhr
 - 03.04 & 04.04: 15:00 – 17:00 Uhr
- Zuteilung auf die Tage und Slots zufällig. Bekanntgabe spätestens Anfang März

Vorbereitung

- Klausur besteht aus ~2 Seiten
- Je Slot gibt es zwei Varianten (A & B)
- 1 Seite Formalien ->
- 1 Seite Aufgaben
 - Wahrscheinlich 4 Aufgaben, von den 3 bearbeitet werden müssen. Bei Bonus nur 2 Aufgaben.

Name: _____ Matrikelnummer: _____

PC: _____ Unterschrift: _____

bearbeitete Aufgaben: ☐ 1 ☐ 2 ☐ 3 ☐ 4

von den Prüfern auszufüllen: ☐ bestanden ☐ nicht bestanden

Leibniz Universität Hannover
Fachgebiet Mensch-Computer-Interaktion

Klausur Programmieren 1
3., März 2018, 8:00 Uhr
Gruppe 01a

Dies ist eine **90**-minütige Klausur. Tragen Sie bitte Ihren Namen, Ihre Matrikelnummer, die PC-Nummer und die bearbeiteten Aufgaben auf diesem Blatt ein und unterschreiben Sie es. Legen Sie bitte Ihren Studierendenausweis und Ihren Personalausweis zur Anwesenheitskontrolle bereit. Es sind keine zusätzlichen Hilfsmittel erlaubt. Schalten Sie bitte Ihr Mobiltelefon aus. Für Notizen können Sie diesen Zettel verwenden. Notizen auf diesem Zettel werden nicht mitbewertet.

Ablauf:

1. Wenn Sie den Bonus erworben haben, bearbeiten Sie zwei der vier Aufgaben (siehe Rückseite), andernfalls bearbeiten Sie drei der vier Aufgaben.
2. Bitte melden Sie sich, sobald Sie fertig sind. Wir werden Ihre Lösung dann überprüfen.
3. Legen Sie diesen Zettel ausgefüllt für uns zur Mitnahme bereit.
4. Nach der Bewertung bitte ausloggen (nicht ausschalten!) und den Raum ruhig verlassen.

Hinweise:

- Loggen Sie sich ein mit Benutzernamen `uni` und Passwort `uni`.
- Öffnen Sie das Terminal über die Menüs `Anwendungen / Hilfsprogramme` und wechseln Sie mit `cd` in das Template-Verzeichnis `Home / Schreibtisch / Klausur01a / group01a`. Listen Sie die Dateien mit `ls`.
- Öffnen Sie das Template-Verzeichnis mit dem Dateimanager per Doppelklick. Öffnen Sie KWrite über `Anwendungen / Zubehör / KWrite`.
- Aktivieren Sie Zeilennummern in KWrite mit `F11`.
- Hinweise zum Kompilieren auf der Kommandozeile finden Sie in den Template-Dateien.
- Ihre Lösungen müssen mit den jeweiligen Testfällen funktionieren und auch generelle Lösungen darstellen.
- Sie dürfen beliebige Hilfsfunktionen implementieren.

Vorbereitung

- ÜBEN!
 - Übungsaufgaben nochmal lösen
 - Eigene Aufgaben ausdenken und lösen
 - Selber eigene Stärken & Defizite herausfinden (Was kann ich, was bereitet mir Probleme?)
- Mit dem Raum, der Konsole sowie dem vorhandenen Editor vertraut machen
 - In F411 an einem PC anmelden und eine Übung dort lösen nur mit dem Editor (kwrite), der Konsole (Terminal) und den Übungsdateien.

ALLGEMEINES

LernLounge

Unterstützung von Studierenden beim Lernen durch studentische Tutorinnen und Tutoren in der InfoLounge

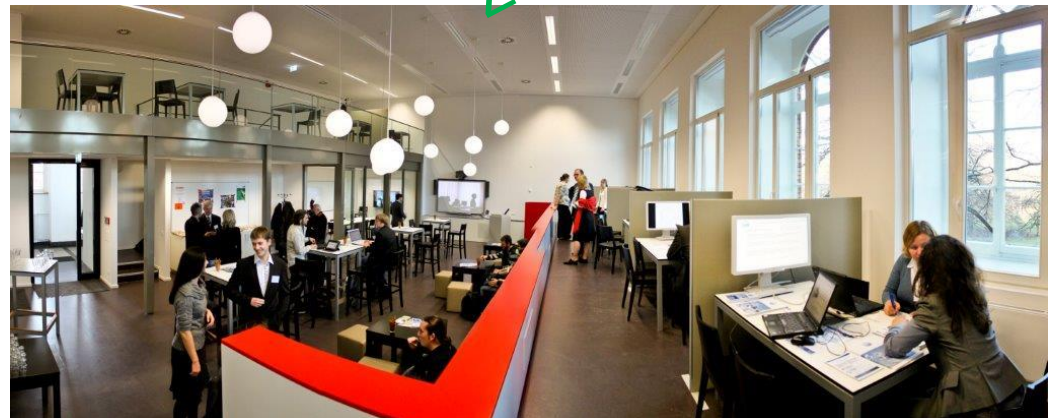
- Die Tutorinnen und Tutoren sind zu festen Zeiten Ansprechperson für fachliche Fragen sowie für Fragen des selbstorganisierten Lernens
- Die Unterstützung erfolgt nach dem Prinzip der minimalen Hilfe

Fächer-Schwerpunkte

- Programmieren
- Grundlagen der Theoretischen Informatik
- Analysis
- Lineare Algebra
- Grundlagen digitaler System
- Elektrotechnische Grundlagen

*Lernlounge auch in der
vorlesungsfreien Zeit!!!
Zeiten ändern sich.*

Start ab Montag den 29.10.18 !!!



Koordination: Isabelle Kross, Tel.: 0511-762 3744, Email: isabelle.kross@et-inf.uni-hannover.de

Where to go for slides, assignments, ...

Stud.IP:

<https://studip.uni-hannover.de>

Go here for slides (e.g., this ones), assignments, and the forum

Handing in assignments:

<https://assignments.hci.uni-hannover.de/WiSe2017/Programmieren1>

Go here to submit your assignments

Online Exam Enrollment with the Prüfungsamt:

<https://www.uni-hannover.de/de/studium/pruefungen/info/informatik/bachelor/termine/>

08.01.2018 – 22.01.2018

Prog1 Lib

More information about the Prog1Lib:

- <http://hci.uni-hannover.de/files/prog1lib/index.html>

Lectures

#	Date	Topic
1	19.10.	Organization, computers, programming, algorithms, PostFix introduction (execution model, IDE, basic operators, booleans, naming things)
2	26.10.	PostFix (primitive types, functions, parameters, local variables, tests), recipe for atomic data
3	2.11.	PostFix (operators, array operations, string operations), recipes for enumerations, intervals, and itemizations
4	9.11.	Recipes for compound and variant data, iteration and recursion, PostFix (loops, association arrays, data definitions)
5	16.11.	C introduction (if, variables, functions), Programming I C library
6	23.11.	Data types, infix expressions, C language (enum, switch, while)
7	30.11.	Compound and variant data, C language (formatted output, loops, struct, union)
8	7.12.	C language (arrays, pointers) arrays: fixed-size collections, linear and binary search
9	14.12.	Dynamic memory (malloc, free), recursion (recursive data, recursive algorithms)
10	21.12.	Linked lists, binary trees, game trees, minimax algorithm
11	11.1.	C language (program structure, scope, lifetime, linkage), function pointers, pointer lists
12	18.1.	Objects, object lists, binary trees, search trees
13	25.1.	Dynamic data structures (stacks, queues, maps, sets), iterators, documentation tools
14	1.2.	This and that, C language (remaining C keywords)

Aufgabe 1a

```
// m maps each team to its result
Map* m = new_map(128, hash_string, strings_equal);
    while (true) {
        String home = s_input(100);
        if (home == NULL || s_length(home) <= 0) {
            s_free(home);
            break;
        }
        String away = s_input(100);
        String goals_home_s = s_input(100);
        int goals_home = i_of_s(goals_home_s);
        s_free(goals_home_s);
        String goals_away_s = s_input(100);
        int goals_away = i_of_s(goals_away_s);
        s_free(goals_away_s);
```

Aufgabe 1a

```
Map* m = new_map(128, hash_string, strings_equal);
while (true) {
    //...
    Stats* s = get_map(m, home);
    if (s == NULL) {
        s = new_stats(0, 0, 0, 0);
        put_map(m, home, s);
    } else {
        s_free(home);
    }
    s->games_played++;
    s->goals_scored += goals_home;
    s->goals_received += goals_away;
    s->score += (goals_home == goals_away) ? 1 :
                ((goals_home > goals_away) ? 3 : 0);
}
```

Aufgabe 1a

```
Map* m = new_map(128, hash_string, strings_equal);
while (true) {
    //...
    s = get_map(m, away);
    if (s == NULL) {
        s = new_stats(0, 0, 0, 0);
        put_map(m, away, s);
    } else {
        s_free(away);
    }
    s->games_played++;
    s->goals_scored += goals_away;
    s->goals_received += goals_home;
    s->score += (goals_home == goals_away) ? 1 :
                ((goals_home < goals_away) ? 3 : 0);
}
```

Aufgabe 1b

```
int size_map(Map* m) {  
    int result = 0;  
    for (int i = 0; i < m->n_buckets; i++) {  
        for (MapNode* kv = m->buckets[i]; kv != NULL; kv = kv->next){  
            result++;  
        }  
    }  
    return result;  
}
```

Aufgabe 1c

```
Entry* entries_map(Map* m) {  
    int n = size_map(m);  
    Entry* entries = xcalloc(n, sizeof(Entry));  
    int j = 0;  
    for (int i = 0; i < m->n_buckets; i++) {  
        for (MapNode* node = m->buckets[i]; node != NULL; node = node->next) {  
            Entry* e = entries + j;  
            e->key = node->entry.key;  
            e->value = node->entry.value;  
            j++;  
        }  
    }  
    return entries;  
}
```

Aufgabe 1d

```
int n = size_map(m);
Entry* entries = entries_map(m);
// sort the entries according to the team scores
qsort(entries, n, sizeof(Entry), compare_teams);
for (int i = 0; i < n; i++) {
    Entry* e = entries + i;
    Stats* s = e->value;
    printf("%s: %d points\n", e->key, s->score);
}
free(entries);
```

Aufgabe 1d

```
int compare_teams(const void* a, const void* b) {  
    const Entry* e1 = a;  
    const Entry* e2 = b;  
    Stats* s1 = (Stats*) e1->value;  
    Stats* s2 = (Stats*) e2->value;  
    if (s1->score < s2->score) return 1;  
    if (s1->score > s2->score) return -1;  
    int goal_diff_1 = s1->goals_scored - s1->goals_received;  
    int goal_diff_2 = s2->goals_scored - s2->goals_received;  
    if (goal_diff_1 < goal_diff_2) return 1;  
    if (goal_diff_1 > goal_diff_2) return -1;  
    if (s1->goals_scored < s2->goals_scored) return 1;  
    if (s1->goals_scored > s2->goals_scored) return -1;  
    return 0;  
}
```


Assignment 14

- [illegible]

Good luck and have fun



`programmieren1@hci.uni-hannover.de`