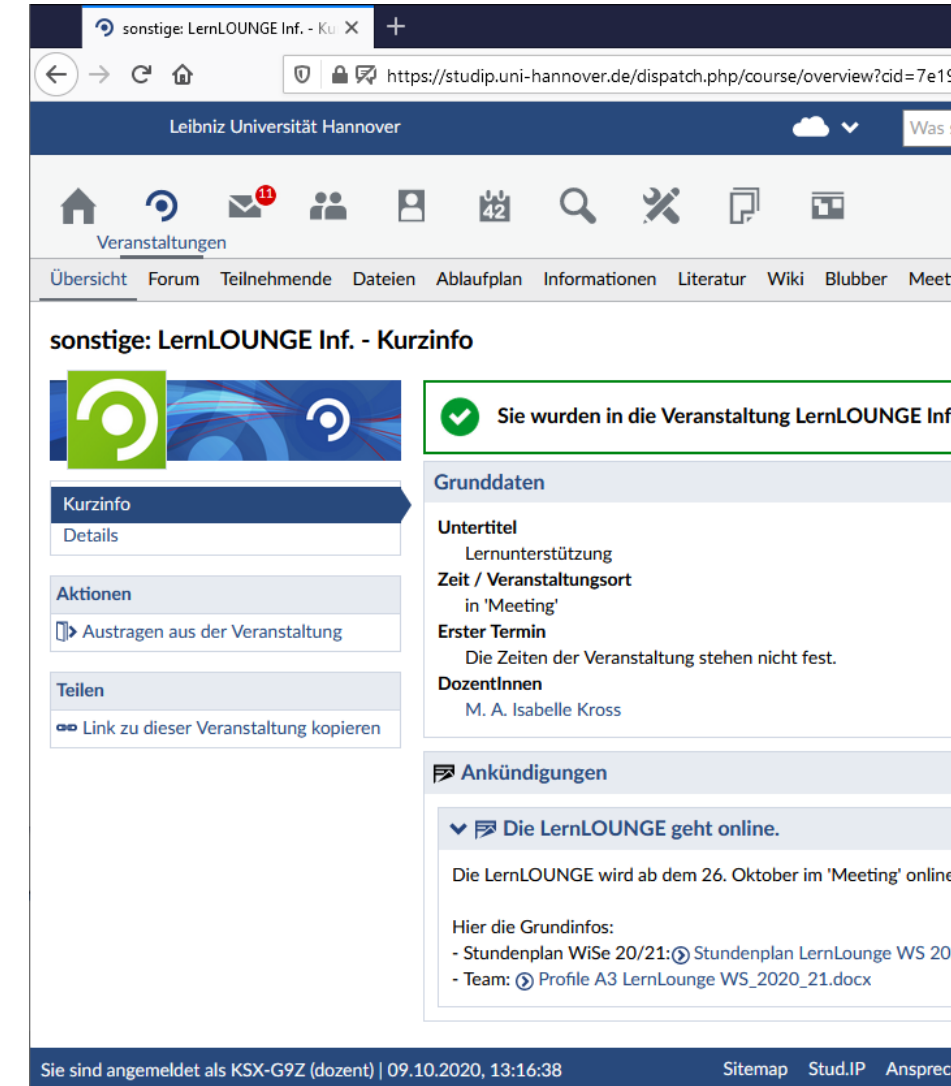


Programmieren 1

Auditorium Exercise 9

LernLounge

- Studentische Tutorinnen und Tutoren helfen Ihnen und unterstützen Sie beim Lernen
- fachliche Fragen, Fragen zum selbstorganisierten Lernen
- Online: <http://go.lu-h.de/jHcTY>
- Fächer-Schwerpunkte
 - Programmieren
 - Analysis, Lineare Algebra
 - Grundlagenfächer 1. und 2. Semester



The screenshot shows the Stud.IP interface for the 'sonstige: LernLOUNGE Inf. - Kurzinfo' page. The top navigation bar includes 'Veranstaltungen' and a search bar. The sidebar on the left contains 'Kurzinfo', 'Details', 'Aktionen', and 'Teilen'. The main content area displays 'Grunddaten' and 'Ankündigungen'. The 'Grunddaten' section includes a green checkmark indicating participation, a title 'Lernunterstützung', a time/location 'in 'Meeting'', and a list of organizers 'M. A. Isabelle Kross'. The 'Ankündigungen' section features a blue announcement 'Die LernLOUNGE geht online.' and a list of links for the syllabus and team profile.

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

LernLounge (ab 26.10.2020)

Zeit		Dienstag	Mittwoch	Donnerstag
10:00 – 10:30		Alexander	Malte	
10:30 - 11:00		Alexander	Malte	
11:00 – 11:30				
11:30 – 12:00				
12:00 – 12:30				
12:30 – 13:00				
13:00 – 13:30			Lennart	
13:30 – 14:00			Lennart	
14:00 – 14:30	Dana		Dana	Lennart
14:30 – 15:00	Dana		Dana	Lennart
15:00 – 15:30	Dana		Dana	Lennart
15:30 – 16:00				Lennart

Schwerpunktfächer

Alexander

- Programmieren
- GTI

Dana (ab 02.11.2020)

- Lineare Algebra
- Analysis

Malte

- GTI
- GDS

Lennart

- Programmieren
- Lineare Algebra

Last Minute Help for your Submission

- Who?
 - Maximilian Lumpe & Lukas Köhler
- When?
 - On thursdays from 6pm to 8pm (Maximilian) & 6 pm to 7pm (Lukas)
- Where?
 - On our Discord Server
- How?
 - Enqueue into the queue of him and wait your turn (like enqueueing for your TA slot)

Questions?

Assignment 8 – Task 2a

```
// (a) todo: implement compare function
int compare(Car car1, Car car2){
    if(car1.year < car2.year){
        return -1;
    }else if(car1.year > car2.year){
        return 1;
    }else{
        return strcmp(car1.brand, car2.brand);
    }
}
```

Assignment 8 – Task 2b

```
// (b) todo: write compare test function void compare_test(void){
    Car car1 = {"VW", 2000, 100, 50000.0};
    Car car2 = {"VW", 1990, 100, 50000.0};
    Car car3 = {"BMW", 1990, 100, 50000.0};
    Car car4 = {"BMW", 1990, 100, 50000.0};
    test_equal_i(compare(car1, car2), 1);
    test_equal_i(compare(car2, car1), -1);
    test_equal_i(compare(car2, car3), 1);
    test_equal_i(compare(car3, car2), -1);
    test_equal_i(compare(car3, car4), 0);
}
```

Assignment 8 – Task 2c

```
// (c) todo: implement sorted function
bool sorted(Car* cars, int length){
    bool sorted = true;
    for(int i = 0; i < (length - 1); i++){
        if(compare(cars[i], cars[i + 1]) > 0){
            sorted = false; }
    }
    return sorted;
}
```


Assignment 8 – Task 2d, e

```
// (d,e) todo: implement random_sort function
int random_sort(Car* cars, int length){
    int swaps = 0;
    while(!sorted(cars, length)){
        int index1 = i_rnd(length);
        int index2 = i_rnd(length);
        Car temp = cars[index1];
        cars[index1] = cars[index2];
        cars[index2] = temp; swaps++;
    }
    return swaps;
}
```

Assignment 8 – Task 2f

```

for( int i = 3; i < 8; i++){
    int swap_sum = 0;
    int number_of_random_cars = i;
    for(int j = 0; j < 100; j++){
        Car* car_park = create_car_park(number_of_random_cars);
        swap_sum += random_sort(car_park, number_of_random_cars);
        delete_car_park(car_park);
    }
    float average_swaps = swap_sum / 100.0f;
    float average_compares = average_swaps * number_of_random_cars;
    printf("Length: %2d\t Swaps: %8.2f, Compares: %9.2f\n", i,
        average_swaps, average_compares );
}

```

Assignment 9

- Will be available on StudIP at about 4 pm
- We will have a brief look inside now

Good luck and have fun!



programmieren1@hci.uni-hannover.de