

---

## DEU DOLMUŞ TAXI

**Please read ALL of the instructions carefully!**

**Due:** 25 November 2018, 11:59 pm

(Time delay is not possible, please do not bid.)

### Goal

In this assignment you are expected to simulate and solve “DEU Dolmuş Taxi” synchronization problem, which its details are given.

On your days at the university, you are waiting for a long time at the entrance of the campus to reach the department or library almost every time. We are launching the DEU Dolmuş Taxi application to facilitate this. This will prevent both all students to create long queues and to move the taxis before be full. Also it will provide a safe and comfortable transportation because taxis will carry students up to the capacity.

With your simulation and solution with mutex and semaphores students and taxis hopefully will no longer wait so long for their courses.

---

## General Requirements

- For this assignment you will work individual.
- The POSIX library (pthread) will be used.
- We compile your code on Debian 8 OS with this code:

```
gcc StudentNumber.c -o StudentNumber -lpthread
```

## Scenario (Implementation Requirements)

- Each taxi has 4 students capacity.
- Students will come to taxi-stop continuously and random periods.
- The taxi stop has 10 taxis and also 10 taxi drivers. Also each taxi contains one taxi driver.
- The states for each taxi:
  - Collect student state: Driver will announce to the students his remaining places to get in the taxi, if it has one or more students in the taxi. For example, each taxi has 2 students and wait two more students, driver call "The last two students, let's get up!"
  - Idle (Empty) state: If there is no student at the taxi stop, the drivers will sleep. The driver does not like to be awake, if not necessary, i.e. no student are waiting.
  - Full and transport state: If there are 4 students in the taxi, the taxi will leave from the taxi stop.

- 
- The states for each student:
    - Waiting in taxi: When a student comes at stop, she/he gets in the nearly full capacity taxi. If no student in taxi, first student wakes driver if he is sleeping. And the students will study or relax in the taxi until the taxi's full.
    - Waiting at taxi stop: If there is no taxi at the taxi stop, she/he's gonna wait for a taxi at the stop.
  - You should do not do the same taxi work constantly so that others can work overtime, so **consider the status of starvation.**

## Submission

Submission will be via Classroom.

- Name your **code file** as: **StudentNumber.c** (do not use another naming standard.)
- You will lose credit for not naming your submission properly. **(15 pts)**
- You **can't** submit your assignment **at late**.

## Honesty

Your submissions will be scanned among each other as well as the Internet repository. Any assignments that are over the similarity threshold of a system for Detecting Software Similarity will get 0. We strongly encourage you not to submit your assignment rather than a dishonest submission.

---

## Grading

- Submit properly (5 pts)
- Documenting the code and coding style (proper indentation, describe critical functions) (15 pts)
- Implementation of main function (Thread creation, join and semaphore initialize) (20 pts)
- Implementation of multithread structure (45 pts)
- Simulate each taxi on the console (15 pts)
  - For example you draw each taxi into a line

T1[#Student] T2[#Student] T3[#Student]... T10[Student]

## For Questions

For any questions about the assignment please send a mail to **okan.ozturkmenoglu@ceng.deu.edu.tr** or **elif.ozturk@ceng.deu.edu.tr** with the subject “**OpSysAssignment1**”. We will try to answer any of your questions as soon as possible, except the ones “Hocam my code does not work, can you fix it” or “I have implemented it but it does not work, can you look at it”. Debuggers are far more suitable options.