INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

**Computer Systems Engineering - I**
Course Assignment 04

Naren Akash, R J
2018111020

# Ober Cab Services

## Compilation and Execution

```
 gcc ober.c -lpthreads -lrt -o ober
./ober < ober.txt
```

## Input

In `ober.txt`,

```
number_of_cabs   number_of_riders   number_of_payment_server
ride_type   maximum_wait_time   ride_time     (for rider 01)
ride_type   maximum_wait_time   ride_time    (for rider 02)
.           .                   .
.           .                   .
.           .                   .
```

```
Note: ride_type is 0 for pool ride and 1 for a premium ride.
```

## The Flow of the Code

1. Semaphores are created and initialized for all the cabs and the payment servers.
2. The Cabs and Servers threads are created and initialized.
3. Whenever a rider arrives, get the required inputs from the user and assign/schedule a cab.
4. If the wait time exceeds the maximum wait time of the rider, he/she exits.
5. After the rider thread finishes the ride for *ride_time* seconds, he/she waits to be assigned to one of the payment servers.
6. After successful payment, the Rider thread exits.
7. Upon finishing all payments, the Server threads exit.

## Reference(s)

The Sleeping Barber Problem in Process Synchronization