EECS 22L: Software Engineering Project in C Language

Lecture 10

Rainer Dömer

doemer@uci.edu

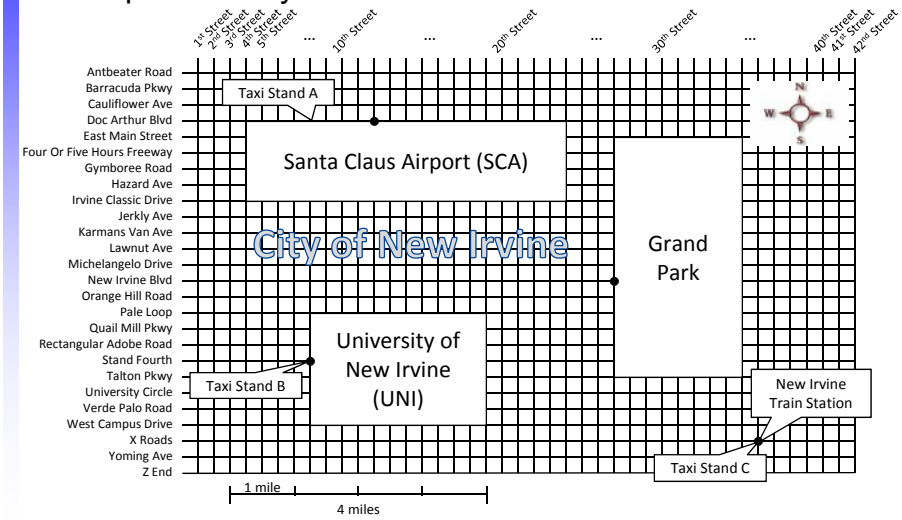
The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 10: Overview

- Project 2 Technical Discussion and Advise
  - Application specification, Updates and Discussion
    - Taxi management settings file
    - Protocol *"Call Taxi"*
    - Protocol *"Taxi Cab"*
    - Pseudo Code for *"Taxi Cab"*
- Towards Object Oriented Programming in C++
  - Introduction to C++ concepts from the C perspective
  - Introduction to classes, objects and strings

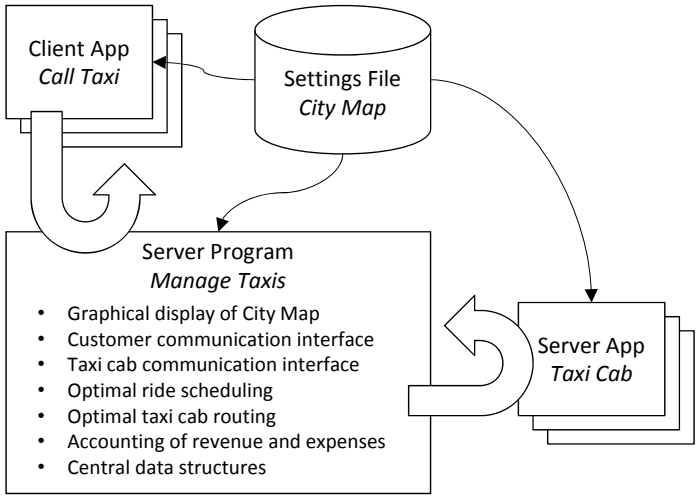
Project 2

- Map of the City of New Irvine



Project 2

- Overall System Specification



Project 2

- System Management Settings (updated)
  - Settings file (extendable)

```
STREETS EAST-WEST:
Antbeater Road, Barracuda Pkwy, Cauliflower Ave, [...], Z End.

STREETS NORTH-SOUTH:
1st Street, 2nd Street, 3rd Street, [...], 42nd Street.

LANDMARKS:
University of New Irvine (UNI) AT Stand Fourth AND 8th Street BOUNDARIES NORTH Pale
Loop EAST 19th Street SOUTH West Campus Drive WEST 8th Street,
[...]
New Irvine Train Station AT X Roads AND 36th Street,
Taxi Stand A AT Doc Arthur Blvd AND 8th Street,
Taxi Stand B AT Stand Fourth AND 8th Street,
Taxi Stand C AT X Roads AND 36th Street.

PARKING:
Taxi Stand A CAPACITY 12 CABS LimoA1(3), LimoA2(3), [...], LimoA6(3), VanA1(6),
VanA2(6), [...], VanA6(6);
Taxi Stand B CAPACITY 12 CABS LimoB1(3), LimoB2(3), [...], LimoB6(3), VanB1(6),
VanB2(6), [...], VanB6(6);
Taxi Stand C CAPACITY 12 CABS LimoC1(3), LimoC2(3), [...], LimoC6(3), VanC1(6),
VanC2(6), [...], VanC6(6).
```


Project 2

- Protocol *"Call Taxi"* (updated)
  - Client Request (in Backus-Naur Form, BNF)

```
<request>  ::= REQUEST <ride>
<ride>     ::= RIDE FROM <location> TO <location> <special>*
<location> ::= CORNER <street name> AND <street name>
             | <landmark name>
<special>  ::= [<time>]
             | [FOR <number> PERSONS]
             | [FIRST CLASS]
<time>     ::= AT <hours>:<minutes>
             | NOW
```

  - Server Response

```
<response> ::= OK BOOKING <code> PICKUP AT <estimated time>
                   COSTS <amount>
             | DECLINED <reason>
             | ERROR <message>
<code>     ::= <unique_reservation_code_or_number>
<amount>   ::= $<dollars>.<cents>
```

Project 2

- Protocol *"Taxi Cab"* (updated)
  - Client Request (in Backus-Naur Form, BNF)

```
<message>   ::= <request>
              | <booking>
              | <instruction>
<request>   ::= REQUEST <taxi> POSITION
<booking>   ::= BOOKING <taxi> <code> <ride>
<instruction> ::= GO <taxi> <direction> <direction>*
              | PICKUP <taxi> <code> <number> PERSONS
              | DROPOFF <taxi> <code> <number> PERSONS
<direction>::= NORTH | EAST | SOUTH | WEST
<taxi>      ::= <type><origin><number>
<type>      ::= "Van" | "Limo"
<origin>    ::= "A" | "B" | "C" | ... | "Z"
<number>    ::= ["1"| ... |"9"]["0"| ... |"9"]*
```

  - Server Response
    - (see next page)

EECS22L: Software Engineering Project in C, Lecture 10 (c) 2015 R. Doemer 7

Project 2

- Protocol *"Taxi Cab"* (updated)
  - Client Request (in Backus-Naur Form, BNF)
    - (see previous page)
  - Server Response

```
<response> ::= OK <taxi> POSITION AT <location>
             | OK <taxi> BOOKING <code>
             | OK <taxi> DRIVING <direction> <direction>*
             | OK <taxi> PICKUP <code>
             | OK <taxi> DROPOFF <code>
             | ERROR POSITION <message>
             | ERROR BOOKING <message>
             | ERROR DIRECTIONS <message>
             | ERROR PICKUP <message>
             | ERROR DROPOFF <message>
             | ERROR <message>
```

EECS22L: Software Engineering Project in C, Lecture 10 (c) 2015 R. Doemer 8

Project 2

- Pseudo Code for *"Taxi Cab"* (proposal)

```
Initialize position at taxi stand (parking), clear directions
Initialize taxi capacity, charges $0, passengers 0, no bookings
Forever:
    Select(timeout 1 simulation time unit)
        On read command:
            If location request, respond position
            If booking request, store booking, respond ok
            If directions, store next moves (overwrite any older ones), respond ok
            If pickup n customers,
                If no matching booking, complain, respond error
                If more customers than room in cab, complain, respond error
                Otherwise pick up customers, respond ok
                Clear directions
            If dropoff n customers,
                If no matching booking, complain, respond error
                Otherwise drop off customers, clear booking, respond ok
                Clear directions
            If invalid, respond error
        On timeout:
            If valid direction, move 1 step, charge $0.20
            If invalid direction, complain, reset directions
            If no direction and at taxi stand (parking), wait
            If no direction and on street, complain
```

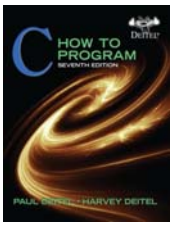

Object Oriented Programming

- Towards Object Oriented Programming in C++
  - C++ can be seen as "improved" C
  - C++ offers a number of new features, including:
    - Inline functions
    - References
    - Default arguments
    - Function and operator overloading
    - Classes and objects
    - Member functions (methods)
    - Constructor and destructor
    - Class and function templates
    - Class inheritance
    - Polymorphism
    - Exception handling

Object Oriented Programming

- "Crash Course" Introduction to C++
  - Selected slides from supplemental text book:

    Paul Deitel, Harvey Deitel,
    "C: How to Program",
    Seventh Edition,
    Prentice Hall, 2013.



  - Excerpts from Chapter 16:
    Introduction to Classes, Objects and Strings