COP290: Design Practices in Computer Science

Vinay Ribeiro

Department of Computer Science & Engg.
IIT Delhi

Course Objectives

- Challenging programming assignments
  - Hone your skill set for the job market

- Design practices
  - Design document
  - Modularity
  - Code Indentation and Commenting
  - Version Control

- Encourage the "start-up" instinct in students
  - Flipkart, Snapdeal, Zomato, Mobiquik, Ola, Housing (by IITans)
  - PM Modi: "On January 16, the government of India will unveil the full Action Plan of Start-up India, Stand-up India. A structure will be presented before you. This programme will be connected to the country's IITs, IIMs, central universities and NITs. Wherever there are youth, they will be linked through 'live connectivity"
  - Infosys founder Narayanamurthy (Jan 6th): "I think all the stars are in place for entrepreneurs and therefore, this is going to be the decade of entrepreneurs"

Teaching Assistants

- Prathmesh Kallurkar (csz128280@cse.iitd.ac.in)
- Megha Gautam (mcs142125@iitd.ac.in)
- Ankit Rohilla (mcs142118@iitd.ac.in)
- Surbhi Jain (mcs142803@cse.iitd.ac.in)
- Preeti Rani (mcs142131@cse.iitd.ac.in)
- Jasmeet Saini (cs5110281@cse.iitd.ac.in)
- Abhishek Bansal (cs5110271@cse.iitd.ac.in)
- Shubham Jindal (cs5110300@cse.iitd.ernet.in)

Course Pre-requisites, Group Size, and Webpage

- Pre-requisite: COL100 (Intro to Computer Science), and COL106 (Data Structures and Algorithms)

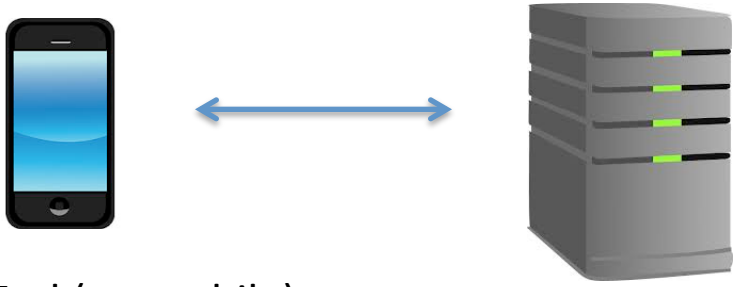
- Group size: 3 (less than 3 only in exceptions)
  - Choose your group carefully
  - Each group should have access to at least one Android device and one laptop/ desktop
  - ALL GROUP MEMBERS MUST CONTRIBUTE, AT DEMOS TAs WILL VERIFY BY ASKING QUESTIONS AND ALSO CHECKING VERSION CONTROL "COMMITS"

- Course webpage available at my homepage: www.cse.iitd.ernet.in/~vinay

- Piazza Discussion group (details also available on webpage)
  - http://piazza.com/iit_delhi/spring2016/cop290
  - Course code is cop290

Assignments

- **Assignment 0**: Simple Android application which registers your group members names on a central server (which will be provided)



FrontEnd (on mobile)

BackEnd (server)

[image src:clker.com]

- **Assignment 1**: Develop front end of a sufficiently complex Android application. Backend provided.
- **Assignment 2**: Develop a complete Android application including the backend.
- **Assignment 3**: To be decided

- Your suggestions for Apps may be incorporated in Assignments

Academic Integrity

- Groups should not share code with each other
  - Moss will be used to detect copying
  - “F” grade for groups copying
  - IITD’s strong policy regarding code copying has benefitted our students in the long run

- You can use some code snippets obtained from Android open source examples on the Internet
  - Do so sparingly
  - Clearly mention the source of the code in comments
  - Understand all the code snippets your borrow from the Internet

Design Document

- Detailed description of what the application does
- User Interface
  - Details of screens visible to user
  - Animations, buttons (enabled/disabled under what conditions)
  - Actions performed when user enters information, presses a button/icon etc.
- High level functions/methods used and descriptions of each one
- Error scenarios and how they are handled
  - E.g. user is asked for Entry number but gives an invalid one
- Use LaTeX to create design document and submit your .tex file(s)

Indentation and Commenting

- Comments describe what the code does at a high-level
  - Not running commentary
- Indentation makes logic clearer
- Use meaningful variable names (not “j”, “k”, etc. which convey nothing to someone reading the code)

```
/**
 * Gets the ith element from the given list by repositioning the specified
 * list listIterator.
 */
private static <T> T get(ListIterator<? extends T> i, int index) {
T obj = null;
    int pos = i.nextIndex();
    if (pos <= index) {
        do {
            obj = i.next();
        } while (pos++ < index);
    } else {
        do {
            obj = i.previous();
        } while (--pos > index);
    }
```

Src: djitz.com

Modularity

- Sub-divide application into smaller parts/modules
- Keep each module as independent of others as possible
  - Modifying one does not affect others
- Keep modules small and in different files
- Allows easier understanding of code, testing and development

Version Control

- Management of changes of software
- Say have stable version 1.0, but want to add some new functionality
- Modify files considerably to get version 2.0, but find this is buggy
- Want to revert to version 1.0 but don't have a copy
- Version Control tools solve the problem
  - Examples: SVN, Git etc.
- Recommend that you use BitBucket
- We will check your "commits" at the time of demo, enter names of those contributing to each commit

Introduction to Android

by

Surbhi and Jasmeet