Welcome to EECS1012!
Net-Centric Introduction to Computing

Amirhossein Chinaei, Winter 2019

ahchinaei@cse.yorku.ca

Office hours: T 9:30-10:30 & W 11:30-12:30 LAS3048

today

- course outline (bird's-eye view)
  - what this course is about
- logistics
  - course organization
  - tests, smq, and mini quizzes
  - optional assignments, grading scheme, *etc.*
- introduction to
  - web application design
    - layering principle, internet/web, HTML/CSS

what is this course about?

- an introduction to computing/programming, via tools and technologies such as:
  - HTML & CCS
  - JavaScript
  - Event-Handling, Test-Driven, & Client-Server Concepts
  - **Computational Thinking**

basic vs. **advanced**?

- this is a **basic** web development course
- assuming **no prior web development skills**

- if you have prior programming experience,
  - you will find the course **too basic**

- if you have no prior experience,
- and do not work hard,
  - you will find the course **too advanced**

course structure

- **lectures**
  - MW 10:30—11:30
  - CLH L
- **labs**
  - R 13:30—16:30 or F 2:00-5:00
  - WSC 106 and 108
  - it's very important you go to your own lab; otherwise 0
- **office hours**
  - T 9:30—10:30 and W 11:30-12:30
  - LAS3048

course structure

- **moodle page**
  - course lectures, lab instructions
  - online quizzes & uploading assignments
  - announcements & discussion forum
  - deadlines and evaluation
  - etc.
- **web resources**
  - we will use many web resources (we do not follow a specific textbook)

evaluation

- in this journey, you have

7 labs & mini-quizzes, 2% each	**14%**
2 in-lab tests, 18% each	**36%**
midterm	**20%**
5 subject-matter quizzes, 1% each	**5%**
final exam	**25%**

letter grade computed using normal mapping

midterm and final

- closed booked
- multiple choice, plus drawing flowcharts
- bring pencils, pens, erasers, and York ID

- midterm
  - Wednesday, Feb 27th, 10:30-11:30
- final exam
  - between April 5th to 20th,
    - *will be determined and announced by the university*
  - length: 2 hours

in-lab tests

- in-lab (bring York ID)
  - test 1: on Feb 14th or 15th depending on your session
  - test 2: on March 28th or 29th depending on your session

- lab tests will require you to write code, **on your own**, under in-lab supervision
- results will be submitted during the lab
- no internet access. "cheat sheets" will be provided

subject-matter quizzes

- 5 different multiple choice tests on ‘key’ subject material relevant to the course
- 20-25 question
- open book/self supervised

labs (each is ~2% of your final grade)

- weekly lab instructions will be available in moodle
- a mini-quiz is available for each lab
  - form Wed at 13:00 to Thu at 12:59pm
  - you write a mini-quiz to demonstrate
    - you have downloaded the instructions prior to the lab,
    - have read it carefully, and done some pre-lab work
- your lab work is graded by TAs during the lab
  - you'll demo it to them, they can ask questions
  - your punctuality also contributes to your grade
  - it's very important you go to your own lab session,
    - otherwise, you receive zero

what would you need to do well?

- **passion, passion, passion**
  - be ready to solve problems, individually
  - be ready to learn details, individually
  - perform a great discussions in lecture, forum, hallway
- pay attention to concepts (in lectures)
- practice the concepts and skills (before labs)
- master your skills by optional assignments (hobby)
- start early the lab works
- lectures and labs are limited
  - yet, for your deep learning, sky's is the limit

let’s start with concepts related to (web application) **design**

principle of layering

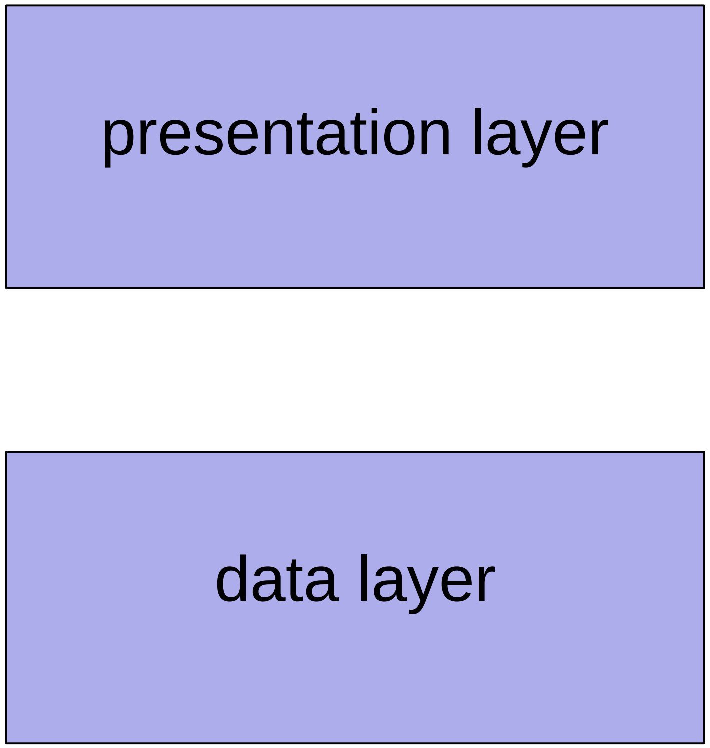
- dividing the application to two+ groups of classes
  - that are functionally or logically related
- such that each layer demonstrates cohesion
- and the dependency among classes is minimized

- **advantages:**
  - modularity, maintainability, reusability

- **disadvantages:**
  - reduced performance

2-layer architecture

- simple application functionality

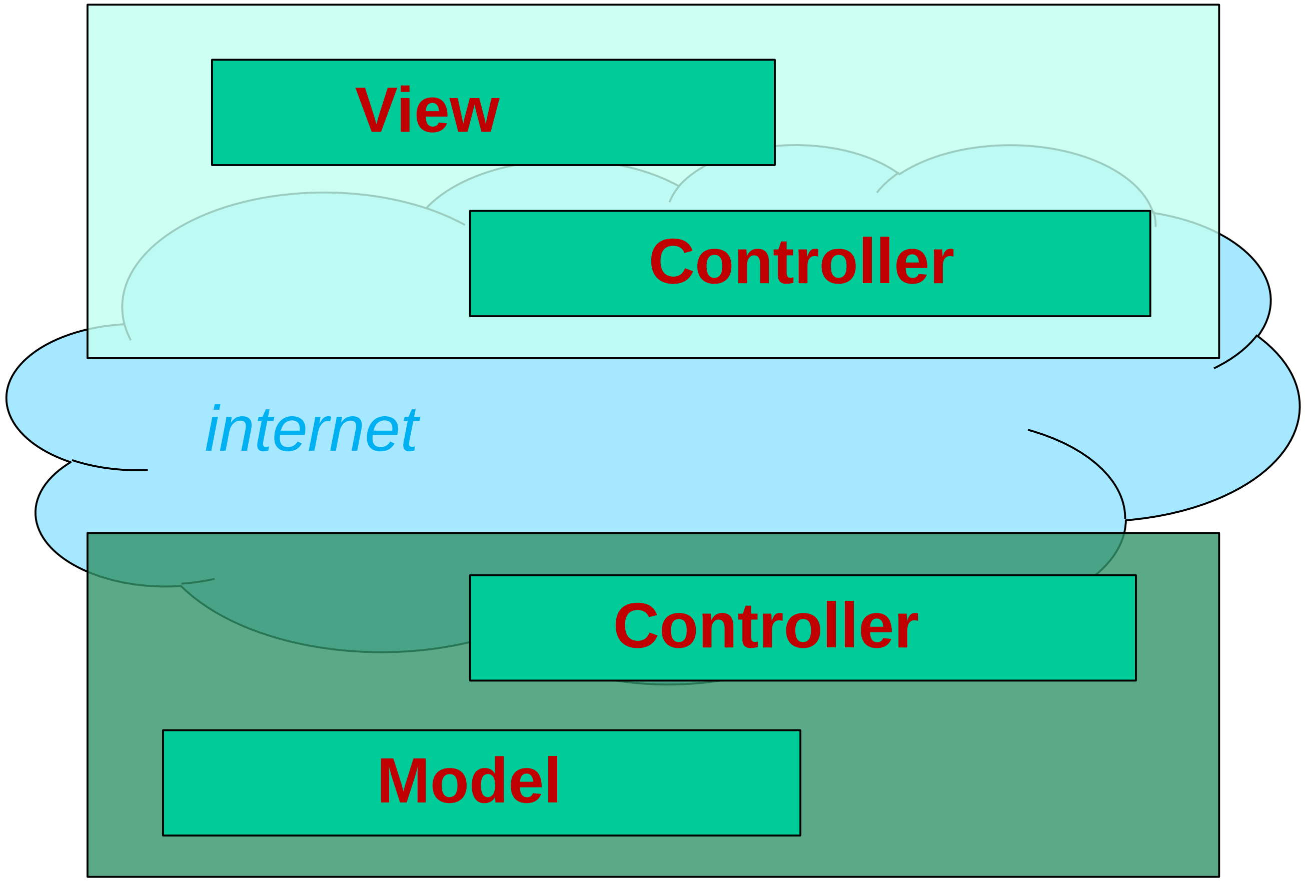


mvc

- the **model** tier
  - represents the **data and logic**
- the **view** tier
  - represents the **user interface**
- the **controller** tier
  - connects and coordinates—**controls**—activities between the view and model

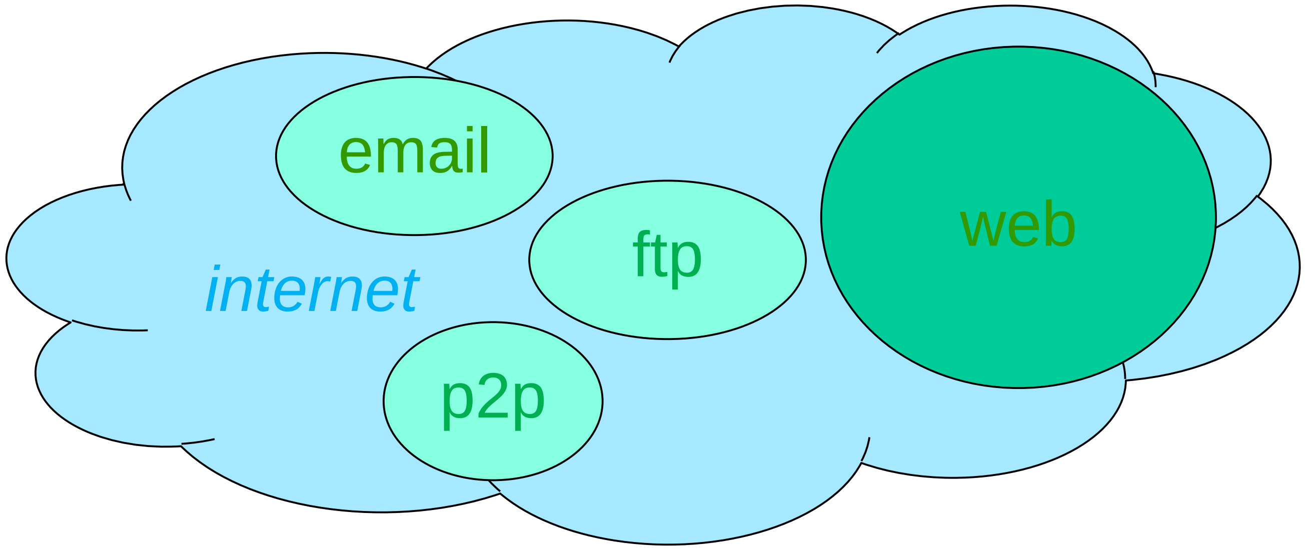
model-view-controller

- **MVC** is a 3-layer pattern



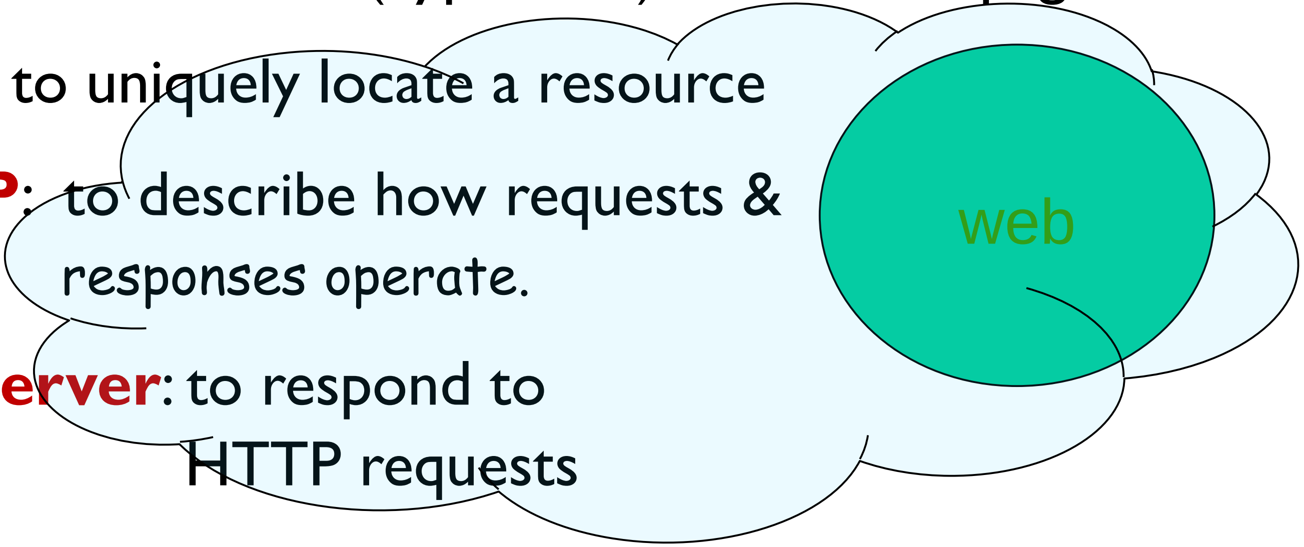
internet & services

❖ **is Internet = WWW ?**



www = web

- it's an information space system—based on request & response—with the following features:
  - **HTML**: to describe (hypertext) documents/pages
  - **URL** : to uniquely locate a resource
  - **HTTP**: to describe how requests & responses operate.
  - **web server**: to respond to HTTP requests
  - **web browser**: to make HTTP requests from URLs and render/display the HTML document received



we start with
<u>html</u>

html

- **HyperText MarkUp Language**

  it's used to describe the **content and structure** of information in a document (web page)

- **general syntax:**

  <element>content</element>

- **example:**

  <h2>1012 is COOOOL</h2>

- **html5** supports multimedia, semantic formatting, cross-mobile applications, and JS APIs.

CSS

- **Cascading Style Sheets**
  - it's used to describe the **appearance** of information
  - it can be embedded in HTML document
    - using the <style> element, or
    - placed in separate .css file
- **example:**

```
h2 {
    color: blue;
    text-align: center;
}
```