ITMD 462/562
Web Site Application Development

# Lecture 1

Fall 2015 – August 26, 2015

# Course Introduction

Web Site Application Development

ITMD 462/562

## Contact

- Professor: Brian Bailey
- Telephone: 312.567.6937
- Email: bbailey4@iit.edu (Email Preferred)
- Office Hours: Before Class and By Appointment

## Lecture Time

- Days: Wednesdays
- Time: 6:25pm to 9:05pm
- Where: Stuart Building, Room 111, IIT Main Campus

## Course Description

Programming the Common Gateway Interface (CGI) for Web pages is introduced with emphasis on creation of interfaces to handle HTML form data. CGI programming is taught in multiple languages. Security of Web sites is covered with an emphasis on controlled access sites. Setup, administration and customization of content management systems including blog and portal sites is introduced. Students design and create a Web site including basic CGI programs with Web interfaces and process data flows from online forms with basic database structures. Prerequisites: [(ITMD 461)] Credit: (2-2-3) (C)

# Course Format

- Lecture Based
- Lectures consist of discussions on the concepts and practical examples
- Slides should not be considered full notes!
  - Take notes during class
  - Don’t tune out and browse the web, you may miss an important concept that is needed for an assignment
- Demos will be shown in class for most topics we discuss.
  - These examples will demonstrate the basics of the topics we discuss. Expect assignments to ask you to go further than the demos in class and require some research on your time.

## Assignment Types

- Two main types
  - smaller lab type assignments expecting to take a small amount of time.
  - Larger multi-week project based assignments
- You will be expected to apply class concepts to solve a simulated problem
- I will deliver a narrative of the problem and a description of what I expect as a deliverable. How you solve the problem has some flexibility.
- Sometimes I will supply some skeleton code or specific libraries to use.
- Graduate students will have extended requirements for each assignment.
- Expect assignment to ask you to do things that may not have completely been covered in class. You will need to take the concepts we discuss and do some additional research to complete the solution.
- Follow the assignment specification document for submission expectations. You will be graded on following those guidelines.
- Assignments are individual (not group) work.

## Knowledge Expectations

- This is a programming course. I expect everyone in this course has been exposed to programming at some level, ideally an object-oriented language.
- I also expect that everyone in the course has at least been exposed to basic database concepts.
- We will discuss object-oriented programming concepts and use basic SQL to work with databases but this is not a full course in either. We will cover the basics but not go into deep detail. You may need to do extra outside research if these topics are not familiar to you.
- You are expected to have a solid knowledge of HTML and CSS since it is a prerequisite to the course.

## Course Policies

## Reading

- There is no required text book, that doesn’t mean there is no required reading.
- Our main resource will be the online php.net official documentation. This will serve as our primary language API guide and reference.
- I will also assign various web resources I find that may be applicable throughout the course.
- I may suggest optional books that would be worthwhile for you to read as additional information and reference.
- If you find interesting articles relative to our class please share.

## Technology Requirement

- This is a programming class. You will need access to a computer and you will need to be able to install software.
    - I have been told the computers in some of the OTS labs will have the software you need. I will check the software list when the new one is published.
- You should be familiar with installing development tools and setting up your development environment. As a developer you would be expected to know how to configure your development environment.
- Any computer operating system will do, Windows, Mac, or Linux. All the work in this class can be done with free software.
- We will discuss software needed and environment setup later.

## Attendance

- If you are registered for the live class you are expected to attend
- Lectures will be recorded and available online. If you miss a class you are expected to watch the lecture online through IIT Online or my screen recording.
- I will be giving occasional quizzes as a factor of attendance and participation
- Online students will get a different timed online quiz, live students are expected to take the quiz in class
  - Sometimes an online quiz may be assigned to both online students and live students.
- There will be discussion forums in blackboard for major class topics and each assignment. Please discuss issues with the class there.
- Quizzes and discussion forum postings are factored into the participation component of the grade.
- See syllabus for detailed policy on these topics.

## Academic Honesty

- Plagiarism and Unauthorized Collaboration
- Do Not Do It. You will receive a Zero for the assignment and be reported for an Academic Honesty violation.
- I am on a lookout for it, it tends to happen every semester to some degree.
- There will be NO EXCEPTIONS to this policy.
- See syllabus for detailed policy
- Plagiarism
  - Copying Code or Words without Attribution
- Unauthorized Collaboration
  - Sharing Code with Classmates
- Your assignments need to be your own code. 80% of your code should be your own. No more than 20% attributed code for the graded component of a project. (primary application libraries not included)

## Grading

- Grading policy is detailed in the syllabus
- The final grade for the class will be calculated as follows:

| | |
|---|---|
| Assignments/Labs | 40% |
| Final Project | 20% |
| Mid-term Exam | 10% |
| Final Exam | 20% |
| Class Participation | 10% |

- Late Policy
  - Detailed in the syllabus grading section

## Syllabus

- The syllabus is my contract with you as to what I will deliver and what I expect from you. If I change the syllabus, I will issue a revised version of the syllabus; the latest version will always be available on Blackboard. Revisions to readings and assignments will be communicated via Blackboard.
- By staying registered in this class you agree to the policies outlined in the syllabus.
- If you ask me a question that is outlined in the syllabus I may answer "It is in the syllabus".
- I will be very firm enforcing the policies in the syllabus.
- Let's review the syllabus.

## Web Technology Review

## HTML

- Hypertext Markup Language
- Proposed by Tim Berners-Lee in 1989 and defined in 1990. HTML, HTTP, URI
- XML like structure to describe the contents of a page in a tree of elements. Originally based on SGML (standard generalized markup language) used within CERN.
- CSS specification added in 1996 to allow for styling and separation of content from presentation in HTML pages.
- We currently should be using the HTML5 specification for all our HTML we do in class.
- *(Demo – Basic HTML Page)*

## URI/URL

- Uniform Resource Identifier
- Uniform Resource Locator
- URI is a unique string that identifies a location of a resource.
- URL typically added the method of locating the resource by adding the protocol component.
- https://danielmiessler.com/study/url_vs_uri/
- http://www.w3.org/TR/uri-clarification/
- https://en.wikipedia.org/wiki/Uniform_resource_locator
- https://en.wikipedia.org/wiki/Uniform_resource_identifier

## HTTP

- Hypertext Transport Protocol
- Protocol for reliable transfer of data from client to server using TCP
- HTTP is the protocol that is used by the browser to send requests and for the server to deliver responses when a web resource is asked for.
- Request – Response model
- HTTPS adds TLS (transport layer security) or SSL (secure sockets layer) encryption to the request/response transfer
- HTTPS created at Netscape in 1994

## HTTP

- HTTP v0.9 – 1991
  - First documented version
- HTTP/1.0 – 1996
  - Officially introduced and recgonized
  - http://www.w3.org/Protocols/HTTP/1.0/spec.html
- HTTP/1.1 - 1999
  - http://www.w3.org/Protocols/rfc2616/rfc2616.html
- 1.0 vs 1.1
  - 1.0 only had GET, POST, HEAD Methods
  - 1.1 requires host header
  - 1.1 adds some cacheing and persistence and more
- http://www2.research.att.com/~bala/papers/h0vh1.html
- HTTP/2 – May 2015 published
  - https://en.wikipedia.org/wiki/HTTP/2

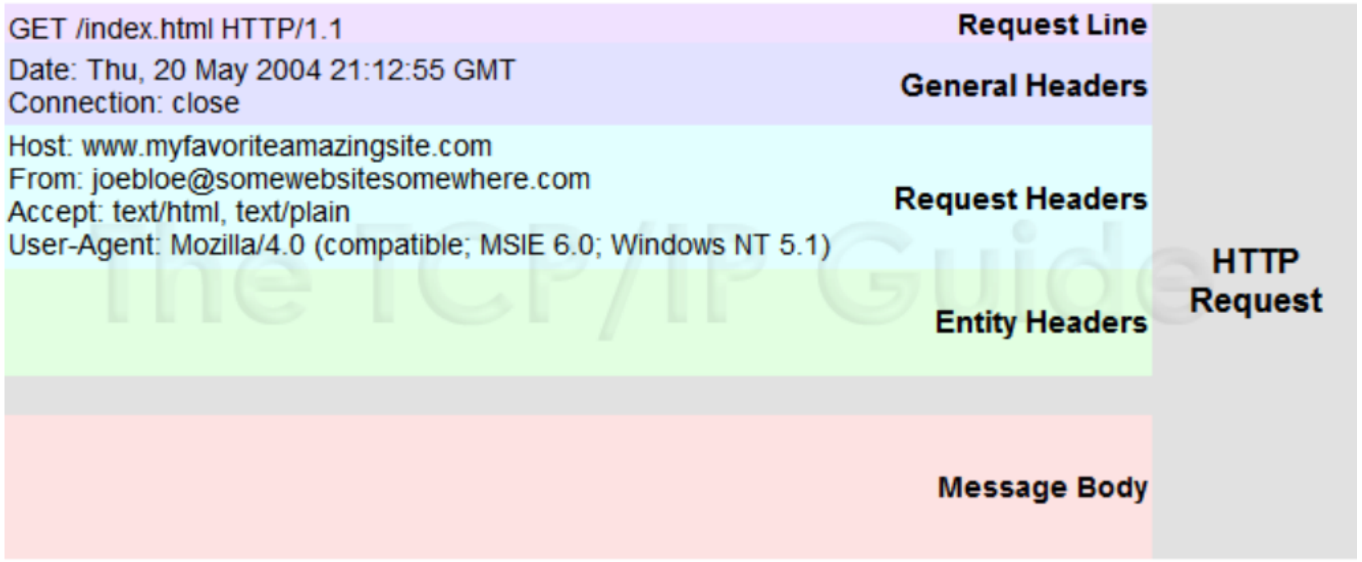
# HTTP Request

- Client Parses the URI
  - protocol://server/request
- Client sends request to Server
  - Usually HTTP protocol

```
[METH] [REQUEST-URI] HTTP/[VER]
[fieldname1]: [field-value]
...
[request body, if any (used for POST and PUT)]
```

- Example - GET /index.html HTTP/1.1

# HTTP Request



```
GET /index.html HTTP/1.1
Date: Thu, 20 May 2004 21:12:55 GMT
Connection: close
Host: www.myfavoriteamazingsite.com
From: joebloe@somewebsitesomewhere.com
Accept: text/html, text/plain
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
```

http://www.tcpipguide.com/free/t_HTTPRequestMessageFormat.htm

# HTTP Methods

- HTTP Methods
  - GET, POST, PUT, DELETE, HEAD, TRACE, CONNECT, OPTIONS
  - First 4 are the common ones. Mostly GET.
  - http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html
- GET
  - Most common, Basically get me this document
  - Any variable or form data is sent as part of the URL
  - http://www.domain.com/?q=232&name=joe
  - Data q=232 and name=joe is available to target page

## HTTP Methods

- POST
  - Second most common method
  - Used often to send form data
  - Any variable or form data is sent in the request body and not appended to the URL
- PUT & DELETE
  - Used mostly with web programming frameworks
  - Used in Ruby on Rails
- HEAD
  - Returns only the Response headers from server
- OPTIONS
  - Used to ask the server for information about the communication options available.

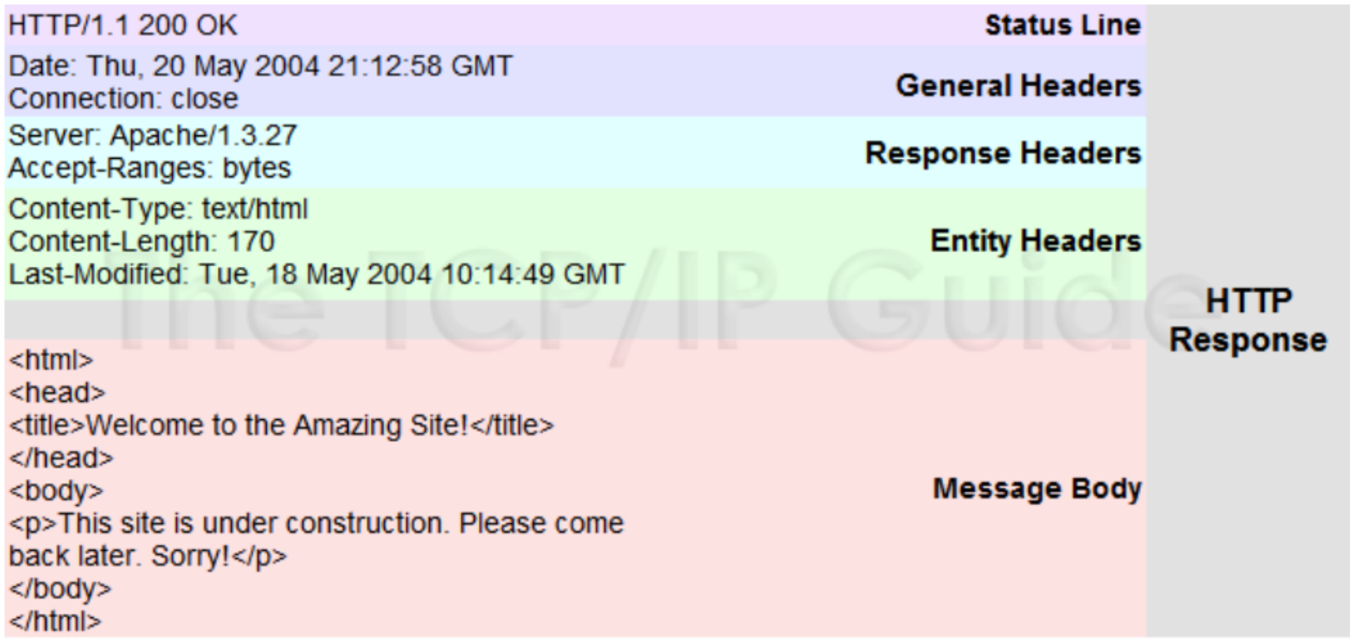
## HTTP Response

- Server sends response to client
  - Usually HTTP Protocol

```
HTTP/[ver] code text
[fieldname1]: [field-value]
...
[response body]
```

- First line is status of request
- Then multiple header fields can follow
- Lastly the response body follows

# HTTP Response



```
HTTP/1.1 200 OK
Date: Thu, 20 May 2004 21:12:58 GMT
Connection: close
Server: Apache/1.3.27
Accept-Ranges: bytes
Content-Type: text/html
Content-Length: 170
Last-Modified: Tue, 18 May 2004 10:14:49 GMT

<html>
<head>
<title>Welcome to the Amazing Site!</title>
</head>
<body>
<p>This site is under construction. Please come
back later. Sorry!</p>
</body>
</html>
```

http://www.tcpipguide.com/free/t_HTTPResponseMessageFormat.htm

## HTTP Response

- First line includes status code. You should know the common codes. Some will be important to our app development.
  - http://www.w3.org/Protocols/rfc2616/rfc2616-sec6.html
- Many headers can be set in both the response and request. We will be setting some of these headers in our web applications.

## Sessions

- A series of requests from one user
- The first request that is not recognized by the server starts a new session and the server considers it a new person.
- Subsequent requests from that person are kept together in the server
- Variables can be stored on the server in the users session
- The session is forcibly cleared by the server

## Dynamic Web Pages and Server Based Web Applications

## Moving beyond static HTML

- Why don't we just write HTML for all our pages?

## Technologies - CGI

- In 1993 at the National Center for Supercomputing Applications (NCSA) a team wrote the first specification for a way of running command line executables in the context of a web request.
- Many other developers used the specification and it became a virtual standard called CGI (common gateway interface)
- A working group was formed in 1997, led by Ken Coar, to get the NCSA CGI definition formally defined and standardized.
- CGI was formally specified as version 1.1 in RFC 3875
  - https://tools.ietf.org/html/rfc3875
- https://en.wikipedia.org/wiki/Common_Gateway_Interface

# Technologies - PHP

- Originally written by Rasmus Lerdorf as a set of CGI scripts in 1994.
- He named the set of scripts "Personal Home Page Tools" often shortened to PHP Tools.
- PHP 3 was a rewrite of the text processor and is the first version that looks similar to today's PHP.
- PHP 3 also got a new name. It was now a recursive acronym "PHP: Hypertext Preprocessor"
- Read the history here
  - http://php.net/manual/en/history.php.php
  - https://en.wikipedia.org/wiki/PHP
- Current released version: 5.6
  - 7.0 is in RC and 6.0 was skipped

# PHP in a webpage

- PHP can be executed various ways these days. The most common way is with an Apache module which executes the code directly in the Apache web server.
- PHP files should be plain text files saved with a .php extension
- The extension triggers its execution in the webserver as php.
- At its most basic it is HTML content with PHP tags added where you want to run PHP code.
- PHP Tags in HTML
  - <?php    //php goes here    ?>
- PHP must run through a configured web server. You can not open a file on your computer directly in your web browser.
- *(Demo – Adding PHP to our HTML Page)*

# Development Environment

- Programming Based Text Editor
  - Notepad++
  - Sublime Text
  - Atom
- https://en.wikipedia.org/wiki/Comparison_of_text_editors
- We will look into PHP IDEs later.
  - Netbeans
  - PHPStorm
  - And others

- Local LAMP Server Package
  - XAMPP
  - MAMP
  - WAMP
- PHP Built in Server
  - Little more involved to install and configure on Windows
  - Built in on the Mac
  - Probably doesn't have all the needed drivers for databases and such out of the box in all environments.

## Assignments

## Reading

- Read the wikipedia page on CGI and the two links in the PHP section.
- Go to http://php.net/manual/en/
    - Read the getting started section
    - Start reading the language reference to get ahead for next week

## Assignment

- No formal assignment
- Try to get your local development environment installed.
  - Programming based text editor
  - Local LAMP stack application
- See if you can get a basic PHP page to be served by the server