Chapter 1: Course Work and Evaluation

Monster

Kobold Lab

Small website psychic, neutral evil

Armor Class 5
Hit Points 20
Speed 1-2 weeks

STR	DEX	CON	INT	WIS	CHA
7 (–2)	15 (+2)	9 (–1)	8 (–1)	7 (–2)	8 (–1)

Senses darkvision 60 ft., passive Perception 8
Languages Python
Challenge 1/8 (25 XP)

Coverage

Everything that is being taught in the lab during the existence of the minion. Including the readings.

Composition

The total weight (or armor class) of the lab component is made up of 9 labs.

Each lab will have several questions that you must answer and upload your code onto your GitHub account and submit the link onto the lab's eclass submission.

Lab 1: Virtualenv & cURL. Introduction to CMPUT 404 labs. Setup virtualenv and understand basic usage of curl.

Lab 2: TCP Proxy. Create a tcp client, proxy server, echo server in Python. Understand how sockets work in relation to web requests. Use multiprocessing for forking new processes.

Lab 3: Common Gateway Interface. Explore the Common Gateway Interface. Refer to the CGI documentation.

Lab 4: Django. Big lab! Build a simple Django website. Understand the fundamentals of Django's MVC architecture using the built in models and views.

Lab 5: Pelican & Basic HTML/CSS. Create a static page using Pelican, or write plain html. Deploy using GitHub pages. Learn to style the basic theme.

Lab 6: Heroku. Deploy the Django application created in Lab 4 to Heroku. Understand the reasoning behind Platform as a Service (PaaS) businesses like Heroku. You may follow the official documentation.

Lab 7: Flask. Create a basic RESTful web application backend using Flask. Consume the API endpoints using cURL and httpie.

Lab 8: Websockets. Learn how to utilize WebSockets and Phaser.io. Create a basic Phaser game with WebSocket connectivity for real-time server to client communication. Use Node.js for our application server. Use TypeScript with Parcel for bundling browser client code.

Lab 9: Authentication. Learn the basics of authentication for web applications. Explore the provided Django Rest Framework applications utilizing HTTP Basic, HTTP Token, and HTTP Session authentication. Understand the high-level intention behind OAuth/OAuth2 and the security implications behind these different authentication schemes.

Archmage Assignment

Medium website psychic, neutral evil

Armor Class 35
Hit Points 35
Speed Varies

STR	DEX	CON	INT	WIS	CHA
10 (+0)	14 (+2)	12 (+1)	20 (+5)	15 (+2)	16 (+3)

Senses —
Languages Python, HTML, CSS, JavaScript
Challenge 12 (8,400 XP)

Coverage

Everything that is being taught in the lecture during the existence of the archmage. Including the readings.

Composition

The total weight (or armor class) of the assignment component is made up of 5 assignments.

Each assignment will have several user stories and requirements that the developer has to implement and upload the code onto your GitHub account and submit the link onto the assingment's eclass submission. Each assignment is worth 7% of the course.

Assignment 1: Webserver. Your task is to build a partially HTTP 1.1 compliant webserver. Your webserver will serve static content from the www directory in the same directory that you start the webserver in.

You are meant to understand the very basics of HTTP by having a hands-on ground up understanding of what it takes to have an HTTP connection.

Assignment 2: Web Client. Your task is to build a partially HTTP 1.1 compliant HTTP Client that can GET and POST to a webserver.

You are meant to understand the very basics of HTTP by having a hands-on ground up understanding of what it takes to have an HTTP connection

Assignment 3: CSS Hell. You will get experience with existing HTML and CSS and you will make some of your own as well.

Assignment 4: AJAX. Your task is to fill in the blanks and connect your browser to your webservice via AJAX. We won't be using XML here unless you want to. I recommend JSON.

Your task is to get this shared drawing program to work using AJAX.

This program allows numerous clients to connect and draw on the same surface and share the drawing they have made with anyone who is currently on.

The goal here is for you to understand the very basics of AJAX, learn a little about canvas, learn a little bit about the overhead of AJAX, learn about JSON and other technologies. Also you'll get to learn about the flask micro-framework.

Assignment 5: Webscokets. Your task is to fill in the blanks and connect your browser to your webservice via Websockets. We won't be using XML here. I recommend JSON.

Your task is to get this shared drawing program to work using Websockets.

This program allows numerous clients to connect and draw on the same surface and share the drawing they have made with anyone who is currently on.

The goal here is for you to understand the very basics of Websockets, learn a little about canvas, learn a little bit about the overhead of Websockets, learn about JSON and other technologies. Also you'll get to learn about the flask micro-framework.

You should be able to see the difference between using websockets and AJAX in this assignment.

Dragon Midterm Champion

Huge website psychic, lawful evil

Armor Class 20
Hit Points 20
Speed 50 minutes

STR	DEX	CON	INT	WIS	CHA
23 (+6)	12 (+1)	21 (+5)	18 (+4)	15 (+2)	17 (+3)

Senses blindsight 60 ft., darkvision 120 ft., passive Perception 22
Languages Python, JavaScript, HTML, CSS
Challenge 15 (13,000 XP)

Coverage

Anything that was covered in the lecture up to the midterm can be tested on the midterm.

Composition

20 multiple choice questions. The midterm is submitted on eclass.

Grade Slayer Tiamat

Gargantuan digital psychic, Lathrain of grades, chaotic evil

Armor Class 33
Hit Points 33
Speed Weeks

STR	DEX	CON	INT	WIS	CHA
30 (+10)	10 (+0)	30 (+10)	26 (+8)	26 (+8)	28 (+9)

Senses darkvision 240 ft., truesight 120 ft.
Languages Python, JavaScript, HTML, CSS
Challenge 30 (155,000 XP)

Description

The web is fundamentally interconnected and peer to peer. There's no really great reason why we should all use facebook.com or google+ or myspace or something like that. If these social networks came up with an API you could probably link between them and use the social network you wanted. Furthermore you might gain some autonomy.

Thus in the spirit of diaspora https://diasporafoundation.org/ we want to build something like diaspora but far far simpler.

This blogging/social network platform will allow the importing of other sources of posts (github, twitter, etc.) as well allow the distributing sharing of posts and content.

An author sitting on one server can aggregate the posts of their friends on other servers.

We are going to go with an inbox model where by you share posts to your friends by sending them your posts. This is similar to activity pub: https://www.w3.org/TR/activitypub/ Activity Pub is great, but too complex for a class project.

We also won't be adding much in the way of encryption or security to this platform. We're keeping it simple and restful.

Choose at least 2 other groups to work with

Composition

Project Parts. Comprised of 3+1 parts

1. Part 0 - Sign up a repo, worth 1%
2. Part 1 - 1/2 way implementation, worth 7%
3. Connect with groups, worth 5%
4. Finalize, worth 20%

RESTful APIs. There are a lot of APIs you must implement for the project, these APIs could be used to implement some of the user stories and requirements.

Take-aways.

- 1 Working website hosted on Heroku or on the lab machines/VM
- 1 GitHub repository
- 1 Presentation
- 1 Video