G54CCS Lab Exercises: Google App Engine (Python)

These exercises will take you through the basics of building simple cloud-hosted applications using the Python programming language on Google's App Engine (GAE) platform. They can all be carried out using the local development server provided as part of the GAE development environment. If you wish to try running them live on GAE you will need to sign up for a Google account on GAE by following the instructions at `http://appspot.com/`.

Lab Exercises

Note that these labs are **not** intended to turn you into expert Python or GAE developers! They are intended to give you insight into the process and problems associated with developing applications for the cloud. You will need to create a new application for each, and enter the code associated with each lab. You should also then attempt any questions/further exercises that are embedded within each lab.

Hello World [pdf]

A simple static "Hello World" page as a GAE application.

Some Python [pdf]

An extension of the first lab to create a more dynamic webpage, and to demonstrate more examples of Python syntax.

A Simple Calculator [pdf]

Moving on from the basic applications of the first two labs, this one introduces more structure to your code, and handles user-supplied parameters, while implementing a simple arithmetic calculator web service.

A Stored Counter [pdf]

This moves on futher, introducing `POST` handling, templates and the use of GAE's simple table-based data store.

Common Problems

As this is the first time this module and these labs have been run, I will collate commonly encountered problems here. Here are what I have so far...

Indentation

In Python, *indentation matters*! Python uses indentation to indicate *block structure*, i.e., which lines of code go together to make up a function or other unit. This is in contrast to languages such as Java, C#, C/C++, which all use curly brackets (`{`,`}`) to do this.

Thus you **must** make sure you have your code correctly indented, otherwise it will not behave the way you expect. Pay particular attention if you use different editors as some use tab characters, some use spaces, and so you can confuse things. If in doubt, use the space bar to insert spaces to indent your code.

Application Names

The GAE development server does not support application names containing UPPER CASE LETTERS.

SDK Location

If you must install the SDK yourself, install it to `C:` not your network-mounted home directory. If you install it on your network-mounted home directory, it will take 15 minutes or more to install, and everything will run very slowly.

Proxy Configuration

Computers within the School are configured to use the School's proxy. If your machine has *not* had an exemption for `localhost` setup, then your browser will attempt to access `localhost` via the proxy, which will fail.

If this happens, simply use the IP address that corresponds to `localhost`, i.e., replace `localhost` with `127.0.0.1`.

Preferred Browser

We recommend using Firefox rather than Internet Explorer on the School's computers, as Internet Explorer hides error messages from you making it much harder to debug your code.

Boilerplate

If you create your application through the GAE launcher, the created application will contain some boilerplate code generated by the launcher. *Pay attention*

to this code too! It must match what is shown in the exercise you are doing — if it does not, you must edit it appropriately.

In general, assume that **all code in your programme is significant**, unless commented out.

Debugging

The *Console Log* is very useful while debugging. If you run your application manually from a Command Prompt, then you will get this by default. If you run your application from within the GAE Launcher, then you will need to explicitly request to see the console log by hitting `CTRL-L` or clicking on the "Logs" button.