Course Outline

The Fine Print
This course is experimental and I reserve the right to change any of the details below, except for the assessment criteria, as the course progresses during the semester. If I think that the assessment criteria needs to change, I will consult the class first.

Aims

This course aims to explore approaches, techniques and tools for building scalable, robust and usable Web services. The course has a substantial practical component, in which students must build a non-trivial Web service within the framework of the arsDigita Community System (ACS).

By the end of the course, you should:

- understand the overall architecture of the Web, and appreciate the strengths and weaknesses of various Web technologies
- be able to design and implement information infrastructures (databases) to support Web services
- be able to connect databases and other informations sources to Web interfaces via server-side scripting languages (e.g. Tcl on AOLserver)
- understand the full range of technologies for building Web-based interfaces (e.g. HTML, DHMTL, CSS, XML, JavaScript)
- appreciate the need for and understand how to construct usable Web-based interfaces
- be able to construct complete Web services (including installation systems, administration manuals and user documentation)

Pre-requisites

You need substantial programming background, in a range of areas of computing. In particular, an understanding of how to build database systems is very useful. Exposure to other systems-based areas such as operating systems and networks is desirable. Also, knowledge of advanced algorithmic techniques from algorithms and artificial intelligence would be helpful.

Web Pages

The course web pages will provide links to all material related to this course. The pages can be accessed via:

From any CSE lab workstation:	file:/home/cs4011/web/index.html
From anywhere on the Internet:	http://www.cse.unsw.edu.au/~cs4011/

You should become familiar with the structure and content of the course web pages early on, and consult them frequently. In particular, all important announcements will be posted to the Course Noticeboard on the opening page.

Staff

Name	Room	Phone	Email	Role
John Shepherd	K17 501-I	9385 6494	jas	Lecturer-in-charge, administration

Syllabus

The syllabus will evolve as the session progresses, but it will certainly deal with at least the following topics:

- web back-end architectures and technologies
- web-server scripting languages (e.g. Tcl)
- web interface technologies (e.g. HTML, JavaScript)
- relational database technology (e.g. SQL, Oracle)
- the ACS toolkit
- designing web interactions, usability
- building scalable, robust, on-line systems
- developing effective on-line communities

Note that, while a number of specific technologies are mentioned in these topics, I ma not planning to spend a lot of time talking about each technology in lectures. You will be largely expected to learn these yourselves. Lectures will simply give some background and look at real-life examples of using the technologies.

Reading

The following book is the recommended text for this course:

- Philip Greenspun, *Philip and Alex's Guide to Web Publishing*, Morgan-Kaufmann, 1999.

The text gives the "philosophical underpinnings" of the course, illustrated by numerous case studies, and contains much sage advice. However, it's not intended to contain all of the technical details (Philip himself describes it as a "the world's first coffee-table computer book"). Most of the technical material for the course is available on-line. For pointers to this and to other useful reading material, see the course Web pages.

Lectures

Each week there will be a 3-4-hour class on Wednesday evening. This will *not* be a regular lecture format with three hours of talk from me. The first two hours (or so) will give an introduction to one component of the syllabus, with copious diversions to Web sites that illustrate various points, and with discussion about the merits of these sites. The remaining time will either be spent in discussing the problem sets and projects, or down in the Lab, where we can investigate the finer points of thevarious technologies. I'll stay until at least 9pm, and later if people have questions down in the Lab.

Note that I am expecting the whole class to actively participate in lecture sessions (one of the luxuries of having a small class). To ensure that people *do* participate, 10% of the final mark for the course will come from my assessment of your contribution to the class during lectures and labs.

There will also be several guest lectures, which will most likely use a more conventional format.

The lecture slots in the last two weeks will be allocated to project presentations/demonstrations.

Project

The major task in this course is the development of a significant, useful Web service using the ACS toolkit. The project is thus worth 60% of the overall mark for the course. Because this is such a large component of the mark, I will be expecting groups to produce a *substantial working* system; accordingly, systems that are too simple or don't work properly will be marked down reasonably harshly (e.g. you can fail this course if your group can't produce a working system).

Project work will be carried out in groups. Each group will work with a *client* and develop a Web service for this client. You are free to look outside the School for projects and clients. If you can't find any project ideas outside the School, I am happy to act as a client either for one of my projects or for a project that the group works out and discusses with me. Note that clients will be asked to provide feedback on what you have developed for them.

Project groups should contain three people, and it is up to each group to organise its own internal dynamics. I will simply assume that every group member contributes equally to the project (if I ask you to document this, this is what you'll tell me anyway). If one person isn't doing their share in the group, it's up to the group to organise remedial action.

Each group will be expected to produce a *substantial working* system, regardless of how many members it contains. I will not be making allowances for groups of size 2 or people who wish to work as individuals. They will be expected to produce a system that is as large as a group containing 3 people. (And, if you haven't got the point, this is really just another way of saying that I insist on groups of size 3).

Project groups will be required to meet regularly with me to discuss the progress of their project.

For more details on how projects will run, specifically for a timetable of deliverables, consult the course web pages. Project groups will be required to deliver at least the following:

- a 5-10 page project proposal
- a 10-minute oral "sales pitch" for their proposal (in lectures)
- a working system, to be demonstrated in class
- a 20-minute presentation on the final product (in lectures)
- "packaging" for the final product, i.e.
  - system overview (5-10 pages)
  - user documentation (15-30 pages)
  - administrator's guide (5-10 pages)
  - configuration/installation scripts

Project groups will be assigned a group account where they can do their development work. Groups must leave the source code of their developing systems accessible to everyone in the class throughout the session. We will conduct occasional code reviews on systems during class.

Problem Sets

The major learning task in this course is to work through the arsDigita Problem Sets. These force you to look at and think creatively about solutions to some typical problems in developing web applications.

Problems sets are to be completed *individually* and submitted using `give`. After you have submitted each problem set, you will be required to discuss your solutions with me. Each problem set contributes 10% towards the overall mark for the course.

Exam

There will be **no** final written exam for this course.

Assessment

The course assessment includes components from the project work, the problem sets, and your contribution during classes.

Component	Type	Percent
Project	Group	~~60%~~70%
Problem Sets	Individual	30%
~~Class Participation~~	~~Individual~~	~~10%~~

Your final mark in this course will be based on a combination of these. Note that you must perform satisfactorily in the individual work before you can pass this course (i.e. the individual work is a hrudle).

The following formula describes precisely how the mark will be computed and how the hurdle will be enforced.

```
pset1  = mark for problem set #1      (out of 10)
pset2  = mark for problem set #2      (out of 10)
pset3  = mark for problem set #3      (out of 10)
~~class  = mark for class participation (out of 10)~~
prop   = mark for project proposal    (out of 10)
demo   = mark for project demo        (out of 10)
packg  = mark for project packaging   (out of 20)
code   = mark for project code        (out of 30)
indiv  = pset1 + pset2 + pset3 ~~+ class~~
group  = prop + demo + packg + code
ok     = mark >= 4/10 for each individual item

final  = indiv + group,  if ok
       = indiv,          otherwise
```

John Shepherd, March 2001