University of Canberra

Computer Engineering 1

Clem Baker-Finch

– 1997 –

0 Unit Organisation

What is this unit about?

- Basic programming skills
- Introduction to software engineering issues
- Basic understanding of computer architecture

It is a *key foundational unit* for Computer Engineering and Computer Science students.

Two sections:

- programming and software engineering (about 80%)
- computer architecture (about 20%)

The *programming* section is a sequence of two parts:

- introductory programming in Haskell (about 10 weeks)
- more advanced programming in Java (remainder of the year)

The *computer architecture* section is one lecture per week for 2nd semester.

Home page

Lecture notes, exercises, assignments and other useful material will be available on the World-Wide Web at the CE1 home page, at:

`http://willow.canberra.edu.au/ clem/CE1/`

Classes and assessment

- Three 50 minute lectures per week.
  You should attend all lectures.
- One 2-hour practical class per week starting in week 2.
  There will be exercises set each week.
- There will be 9 assignments set throughout the year.
  (3 assignments in the Haskell part.)
- There will be an exam at the end of semester 1 and two exams at the end of the year.

How we judge a PASS grade:

- A satisfactory standard in all exams *AND*
- A satisfactory standard in assignments
  (you must submit solutions to *all* assignments) *AND*
- A satisfactory attendance and performance in practical classes.

How *YOU* can achieve a PASS (or better):

- Attend lectures
- Read the textbook
- Review your notes and the textbook
- Experiment — write small programs to test ideas and techniques
- Complete the assignments
- Start work on assignments EARLY
- Attend all pracs *and* make good use of your time

This is a *PRACTICAL* subject, so you *must*

practice and experiment, practice and experiment, practice and experiment, practice and experiment, practice and experiment . . .

Previous programming experience?

If you have had experience with a language such as Pascal or C it will probably *not* be much advantage for the first part of the course (Haskell is a *functional* programming language).

If you have no programming experience, Haskell is an easy way to get started.

If you find it too easy (?) take the opportunity to delve deeper. I plan to set up some (optional) more advanced challenge exercises. Watch the CE1 home page.