CITS 3242
Programming Paradigms

Lecture 1:
Introduction and Motivation
Quote

- A language that doesn't affect the way you think about programming is not worth knowing.

  -- Alan Perlis (Turing Awardee, 1966)
What is this unit about?

- What exactly is a programming paradigm?
  - A paradigm is a “philosophical or theoretical framework of any kind” (Merriam-Webster).
- Object-orientation is a paradigm you should all be familiar with.
- This unit focuses on alternative paradigms.
  - Including comparisons and interactions with mainstream paradigms.
What paradigms will we see?

- **Functional Programming**
  - Based on defining mathematical functions.

- **Concurrent Programming**
  - Allows many things to happen concurrently.

- **Logic Programming**
  - Based on logical deduction from a set of rules.
Languages

- Programming languages are generally designed with one paradigm in mind.
  - E.g., Java is designed for object-orientation.
- We’ll use two languages in this unit:
  - Microsoft’s functional/multiparadigm language F#, which supports concurrency and objects.
  - The logic programming language Prolog (briefly).
Why learn alternative paradigms & languages?

- Because they make different tradeoffs, and so can make programming easier sometimes.
  - Also, for a deeper understanding of the mainstream.

- Because they may become mainstream.
  - Object orientation was once “alternative”.

- Because they influence the mainstream.
  - E.g., garbage collection from functional languages (and much more).
Why learn alt. paradigms & languages?

- **Because it will broaden the way you think about programming, and help you solve problems in new ways.**

- **Sapir-Whorf Hypothesis:** (linguistics) *We dissect nature along lines laid down by our native language*

- **Boole:** *Language is an instrument of human reason, not merely a medium for the expression of thought* [An Investigation of the Laws of Thought, 1854]

- **Orwell:** *The purpose of Newspeak was not only to provide a medium of expression for the world-view and mental habits proper to the devotees of Ingsoc, but to make all other modes of thought impossible* [1984, 1949]
Why learn alternative paradigms & languages?

- Because it will prepare you for future paradigms & languages.
- Because it will help you in comparing languages.
- Because it will help you understand languages at a deeper level, instead of just the syntax or sales pitch.
Why Functional Programming?

- Functional programming is one of the oldest paradigms.
  - Lisp originated in 1958!
  - It is still widely used, and has been highly influential.
- Functional programming can be very elegant, and has a strong mathematical foundation.
- It is the “next big thing”.
- Many other paradigms can be neatly interpreted in terms of functional programming.
Why Concurrent Programming?

- Because concurrency is required by the real world.
- Because multi-core programming requires concurrency.
- Because concurrent programming is difficult to get right.
- Because comparing concurrent paradigms shows just how different solutions to a problem can be.
- Because functional programming provides solutions to some issues in concurrency.
Why Logic Programming?

- Because it has a long history.
- Because it allows certain things that are difficult in other paradigms.
- Because it is a very different paradigm to the mainstream.
- So that we can relate it to functional programming.