CITS1001
OBJECT ORIENTED
PROGRAMMING AND
SOFTWARE ENGINEERING
In CITS1001 you will learn how to write simple computer programs using the Java programming language.

The purpose of today’s lecture is to give an overview of the unit and what you will have learnt by the end.
What is a computer program?

• A sequence of elementary instructions that when followed achieves a specified goal

- a procedure, a recipe, an algorithm

2. Stir fry 1 clove garlic, 1 tbsp black beans, 100g preferred roast meat.
3. Add 2 small bok-choy and toss to coat.
4. Scatter a pinch of sugar and add 2 tbsp water.
5. Reduce heat, cover and cook for two minutes.

Stephanie Alexander, the cook’s companion
What is a programming language?

• A program for a computer to follow must be expressed completely unambiguously
• There are many different programming languages in which programs can be written
• In order to write a working program, you need to learn
  • the vocabulary and syntax of the language, so that you can write statements that can be understood
  • how to make sequences of legal statements that do simple tasks
  • how to express what you wish the computer to do in a simple enough
  • way to translate into the programming language
• Similar to learning the words, how to form sentences and how to write a story when learning a new human language
Programming in the large

Most computer programs these days

- run continuously
- respond to input from users and other programs
- deliver output to users and other programs
- are programmed by large teams of programmers
- are too big and complex for any one person to understand
For example, a web application

- responds to mouse clicks, and text entries
- communicates with computers across the Internet
- formats and displays Web pages
- starts up “helper applications” if necessary
Complexity

How is something moderately complex organized in the real world? Consider a *restaurant* ...

- Diners want meals
- Chefs prepare dishes
- Waiters take orders from diners, and bring food to the tables
- Busboys collect and wash plates
- Barmen prepare and serve drinks
- The maitre’d makes reservations and seats diners

*Each type of person provides a narrow range of services. The restaurant operates through the co-operative interaction of all the restaurant staff*
Object Oriented Programming

- OOP refers to the process of designing and implementing a co-operative community of interacting objects.
  - each object provides a small number of relatively simple services
  - objects communicate with each other to exchange information
  - complex behaviour is achieved by the co-operation of objects

- Using objects allows us to focus on a problem at a high level without worrying about the minor details.

- That is, we use ABSTRACTION to manage COMPLEXITY
Object Oriented Programming
What does OOP provide?

• Complexity management
  • *encapsulation* and *information hiding* work on the principle that objects should interact *only* through pre-specified *interfaces*
  • software development can be more reliably divided between independent groups

• Reusable software
  • *class libraries* provide easy access to many standard services
  • developing *software components* that match the reliability and interchangeability of hardware components is an elusive goal

• Natural modelling
  • problem identification, program design and program implementation all follow same process
An early algorithm
1800-1600 BC Mesopotamia

A cistern. The height is 3,20, and a volume of 27,46,40 has been excavated. The length exceeds the width by 50. You should take the reciprocal of the height, 3,20, obtaining 18. Multiply this by the volume, 27,46,40, obtaining 8,20. Take half of 50 and square it, obtaining 10,25. Add 8,20 and you get 8.30,25. The square root is 2,55. Make two copies of this, adding to the one and subtracting from the other. You find that 3,20 is the length and 2,30 is the width. This is the procedure.

Source: Ancient Babylonian Algorithms, D.K. Knuth, CACM 1972
(see cits1001 resources page)
The first computer programmer

• In 1837 Charles Babbage proposed a type of computer called the Analytical Engine
• In 1842 Ada Lovelace wrote the first algorithm for computing Bernoulli numbers using the Analytical Engine
• A Babbage Engine was not actually built until 2002

Augusta Ada King, Countess of Lovelace (10 December 1815 – 27 November 1852)
Source: http://en.wikipedia.org/
Are there any problems we can’t solve with a computer program?

- **YES**
- Example, the **halting problem**: there can be no program to decide whether another program will halt or run forever on a given input.
- Alan Turing 1937

- And furthermore, some things are computable but would take unreasonable time or space.
- Example, the **travelling salesman problem**: given a list of cities and distances between each pair, is there a route with length less than k that visits all cities?
  (can I visit all my friends with half a tank of petrol?)
That is …

"You can never find general mechanical means for predicting the acts of computing machines; it’s something that cannot be done … ”

From "Snooping the Loop Snooper"
http://www.lel.ed.ac.uk/~gpullum/loopsnoop.html
Can computers think?

- Turing test 1950s: can a computer program show “intelligence”?
  
  Alan Turing, Mind, October 1950

- Can a computer “think”?

- Researchers in Artificial Intelligence are still studying this question today

Alan Mathison Turing
Source: en.wikipedia.org
A first programming course

• Becoming an expert programmer (or expert anything) requires at least 10,000 hours

• In your first 150 hours in CITS1001 you will learn
  • What are programs for?
  • Syntax – speak the language
  • Semantics – understand what is done
  • Algorithms – designs and plans
  • Tools - development, check, test, debug
  • Quality Assurance – good vs bad programs
By mid-semester you will be able to

- Program a drawing application to create and modify image collages
- You will be manipulating collections of objects
By the end of the unit

You will be able to manipulate more complex data structures and design algorithms to solve problems.

For example, program a (seemingly) intelligent chat program.
Recommended Reading

See the resources page on the CITS1001 web pages for more links