Two classes in one

There are two classes with a common lecture stream:

- COMP1400
- INFS1609

Classes have separate labs and assessment tasks.

COMP1400

Lecturer in Charge: Malcolm Ryan
Email: cs1400@cse.unsw.edu.au
Subject admin: ???
Tutor: ???

INFS1609

Lecturer in Charge: Claude Sammut
Email: is1609@cse.unsw.edu.au
Subject admin: ???
Tutor: ???
Class announcements

Announcements will be made in lectures and posted on the class web pages.
It is your responsibility to keep up to date with announcements.

Labs

Programming can only be learned by doing.

Lab exercises will be posted on the web pages for each subject.

COMPI400 labs: CHI Lab on the ground floor of the K17 Building (CSE).

INFS1609 labs: Quad Lab 7
Labs start in Week 2.

COMPI400 Assessment

COMPI400 will be assessed based on:
- Lab mark (10%)
- Three assignment tasks (10% each)
- A practical exam in Week 13 (30%)
- A written exam (30%)

INFS1609 Assessment

INFS1609 will be assessed based on:
- Lab mark (10%)
- Three assignment tasks (10% each)
- A written exam (60%)
Java and BlueJ

This course will be taught using the Java programming language and the BlueJ interactive development environment.

Java and BlueJ can be downloaded for free at: http://www.bluej.org/

Textbook

Objects First with Java:
A Practical Introduction using BlueJ
David J. Barnes & Michael Kölling
Pearson, 5th edition 2012
http://www.bluej.org/objects-first

What is a computer?

• It is an incomplete machine
  • computer hardware only provides the building blocks for a working machine
• A program assembles the blocks and says how they will be used and in what order

What’s the difference?
A program is a sequence of instructions that make a machine

private static int min(int[] nums) {
    int m = nums[0];
    for (int i = 1; i < nums.length; i++) {
        int n = nums[i];
        if (n < m)
            m = n;
    }
    return m;
}

What does this say?

ΣΟΚΡΑΤΗΣ: κατάδην χθές είς Παιαν ἐμέτα
Γλαύκωνος τοῦ Ἀρίστονος προσευχόμενος τη τῇ ἡμερήσιᾳ
καὶ ἁμα τὴν ἀρετὴν βουλόμενως θαύμασθαι τίνα
πρότειν ποιήσωσιν ἢ τὸ πρῶτον ἄγαντος, καλή μὲν
μάλλον ἄστεγον ἡφαίνετο πρότειν ἢν οἱ Θεῖαι ἐπιμπούν,
προσευχόμενοι δὲ καὶ ἔκλεισον τοῦ πρὸς τὸ ἄστυ,
κατιδῶν οὕνεκα πάντως ἡμᾶς ὑπερανέμονον
Πολέμαρχος ὁ Καρνίγλων ἔκλεισεν ἐκείνον τὸν παῖδα
περιομέναι ἢ καλεῖσαι, καὶ μου ὑπισθεὶν ὁ παῖς
παλαβόμενος τῷ ἂνακτῷ, καλεῖς ὑμᾶς, ἢφη,
Πολέμαρχος περιομέναι.
...is the same as this...

```python
def min(nums):
    m = nums[0]
    for i in range(0, length(nums)):
        if (nums[i] < m):
            m = nums[i]
    return m
```

...is the same as this...

```python
def min(nums):
    m = nums[0]
    for i in range(0, length(nums)):
        if (nums[i] < m):
            m = nums[i]
    return m
```

...and this...

```prolog
min([X], X).
min([X | Xs], X) :-
    min(Xs, Y),
    X < Y, !.
min([_ | Xs], Y) :-
    min(Xs, Y).
```

...is the same as this...

```python
def min(nums):
    m = nums[0]
    for i in range(0, length(nums)):
        if (nums[i] < m):
            m = nums[i]
    return m
```

...and this...

```prolog
min([X], X).
min([X | Xs], X) :-
    min(Xs, Y),
    X < Y, !.
min([_ | Xs], Y) :-
    min(Xs, Y).
```

Procedural thinking

Programming is about thinking in terms of step-by-step procedures to achieve an outcome.

Mastering procedural thinking is more important than any particular programming language.
Programming is like Pancakes

1. Put ingredients in bowl.
2. Beat until smooth.
3. Heat frying pan.
4. Pour 1/4 cup of mixture into pan.
5. Wait for bubbles.
6. Flip.
7. Wait 30s or until brown underneath.
8. Transfer to plate.
9. Repeat steps 4-8 until all mixture is used.

Computers are stupid

To put ingredients in bowl:
1. Take bag of flour from cupboard
2. Take bottle of milk from fridge
3. Take carton of eggs from fridge
4. For each person served:
   1. Transfer 1 cup of flour from bag to bowl
   2. Transfer 1 cup of milk from bottle to bowl
   3. Remove 1 egg from carton
   4. Crack egg and pour contents into bowl
   5. Put eggshell in compost bin

Computers are picky

```java
void addIngredients(Bowl bowl, int numberOfPeople)
{
    Flour flour = theCupboard.get("Flour");
    Milk milk = theFridge.get("Milk");
    List<Egg> eggs = theFridge.get("Eggs");

    for (int i = 0; i < numberOfPeople; i++)
    {
        flour.transfer(1, bowl);
        milk.transfer(1, bowl);
        Egg egg = eggs.remove(0);
        egg.crackInto(bowl);
        egg.dispose();
    }
}
```

The 6 Stages of Programming

1. Requirements What do they want?
2. Specification What should it do?
3. Design How will it work?
4. Implementation How is it made?
5. Testing Does it actually work?
6. Debugging What went wrong? How to fix?
The essential skills of programming

Procedural thinking: knowing how to write a sequence of instructions to achieve an outcome.

Language literacy: knowing how to read and write code.

Testing and debugging: knowing how to test your code and track down bugs.

The 7th Stage of Programming

1. Requirements
2. Specification
3. Design
4. Implementation
5. Testing
6. Debugging
7. Documentation

Good code

A well-written computer program:

1. Tells the computer what to do,
2. Tells another human being what the computer is going to do.

Bad Style

```csharp
void AddIng(Bowl b, int n) {
    Flour f = c.Get("Flour");
    Milk m = fr.Get("Milk");
    List<Egg> e = fr.Get("Eggs");
    for (int i=0; i<n; i++)
    {
        f.Put(1,b); m.Put(1,b);
        Egg e1 = e.rem(0);
        e1.cr(); e1.Put(b);
        e1.Disp();
    }
}
// Add pancake ingredients to bowl:
// bowl - the bowl to put them in
// numberOfPeople - the number of people to serve
// Makes 2-3 pancakes each

void AddIngredients(Bowl bowl, int numberOfPeople)
{
    // 1 cup of SR flour per person
    Flour flour = theCupboard.get("Flour");
    flour.transfer(numberOfPeople, bowl);

    // 1 cup of milk per person
    Milk milk = theFridge.get("Milk");
    milk.transfer(numberOfPeople, bowl);

    // 1 egg per person
    List<Egg> eggs = theFridge.Get("Eggs");
    for (int i = 0; i < numberOfPeople; i++)
    {
        Egg egg = eggs.remove(0);
        egg.crackInto(bowl);
        egg.dispose();
    }
}