

COMP 1400

Programming for Designers

INFS 1609

Business Programming

Malcolm Ryan
Claude Sammut

Two classes in one

There are two classes with a common lecture stream:

- COMPI400
- INFSI609

Classes have **separate labs** and **assessment tasks**.

COMP1400

Lecturer in Charge: Malcolm Ryan

Web pages: <http://blogs.unsw.edu.au/comp1400/>

Email: cs1400@cse.unsw.edu.au

Subject admin: Tim Wiley

Tutor: Sim Mautner

INFS I 609

Lecturer in Charge: Claude Sammut

Web pages: <http://blogs.unsw.edu.au/infs I 609/>

Email: is I 609@cse.unsw.edu.au

Subject admin: Tim Wiley

Tutor: Tim Wiley

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)**.

INFSI609 labs: ????

Lecture videos

Lectures will be video recorded and uploaded to [UNSW TV](#).

Videos are **not a substitute** to attending lectures.

No special consideration will be given if you miss important announcements or lecture material because you didn't attend.

COMPI400

Assessment

COMPI400 will be assessed based on:

- Three assignment tasks (10% each)
- A practical exam in Week 13 (40%)
- A written exam (30%)

INFS I 609

Assessment

INFS I 400 will be assessed based on:

- ...

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

Prentice Hall / Pearson Education, 2008

<http://www.bluej.org/objects-first/>

What does this do?

```
private static int min(List<Integer> nums) {  
    int m = nums.get(0);  
    for (Iterator<Integer> it = nums.iterator(); it.hasNext();) {  
        int n = it.next();  
        if (n < m) {  
            m = n;  
        }  
    }  
    return m;  
}
```

What does this say?

ΣΩΚΡΑΤΗΣ· κατέβην χθὲς εἰς Πειραιᾶ μετὰ
Γλαύκωνος τοῦ Ἀρίστωνος προσευξόμενός τε τῇ θεῷ
καὶ ἅμα τὴν ἐορτὴν βουλόμενος θεάσασθαι τίνα
τρόπον ποιήσουσιν ἅτε νῦν πρῶτον ἄγοντες. καλὴ μὲν
μέντοι ἦττον ἐφαίνετο πρέπειν ἢν οἱ Θοῤᾳκες ἔπεμπον.
προσευξάμενοι δὲ καὶ θεωρήσαντες ἀπῆμεν πρὸς τὸ
ἄστυ. κατιδὼν οὖν πόρρωθεν ἡμᾶς οἴκαδε ὠρμημένους
Πολέμαρχος ὁ Κεφάλου ἐκέλευσε δραμόντα τὸν παῖδα
περιμεῖναί ἐ κελεῦσαι. καί μου ὀπισθεν ὁ παῖς
λαβόμενος τοῦ ἱματίου, κελεύει ὑμᾶς, ἔφη,
Πολέμαρχος περιμεῖναι.

This...

```
private static int min(List<Integer> nums) {  
    int m = nums.get(0);  
    for (Iterator<Integer> it = nums.iterator(); it.hasNext();) {  
        int n = it.next();  
        if (n < m) {  
            m = n;  
        }  
    }  
    return m;  
}
```

...is the same as this...

```
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...

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

...and this...

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

...is the same as this...

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

...and this...

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

...and this.

```
LDA $0,X  
LOOP: CMP $0,X  
      BCC SKIP  
      LDA $0,X  
SKIP: DEX  
      BNE LOOP  
      RTS
```

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**

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

```
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

The 6 Stages of Programming

I. Requirements

The 6 Stages of Programming

1. Requirements
2. Specification

The 6 Stages of Programming

1. Requirements
2. Specification
3. Design

The 6 Stages of Programming

1. Requirements
2. Specification
3. Design
4. Implementation

The 6 Stages of Programming

1. Requirements
2. Specification
3. Design
4. Implementation
5. Testing

The 6 Stages of Programming

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

The 6 Stages of Programming

1. Requirements **What do they want?**
2. Specification
3. Design
4. Implementation
5. Testing
6. Debugging

The 6 Stages of Programming

1. Requirements **What do they want?**
2. Specification **What should it do?**
3. Design
4. Implementation
5. Testing
6. Debugging

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
5. Testing
6. Debugging

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 **What makes it tick?**
5. Testing
6. Debugging

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 **What makes it tick?**
5. Testing **Does it actually work?**
6. Debugging

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 **What makes it tick?**
5. Testing **Does it actually work?**
6. Debugging **What went wrong? How to fix?**

The 7 Stages of Programming

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

The 7 Stages of Programming

1. Requirements

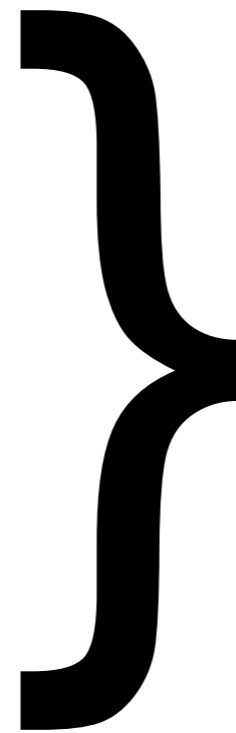
2. Specification

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

```
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();
    }
}
```

Good style

```
// 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();
    }
}
```

Style Guide

[http://www.bluej.org/objects-first/
styleguide.html](http://www.bluej.org/objects-first/styleguide.html)