Programming - and in general software development - is not a purely theoretical skill, nor is it something you can do well without learning some fundamental concepts. Unfortunately, far too often, teaching fails to maintain a balance between theory/principles and practicalities/techniques.

Consequently, we see people who basically despise programming (“mere coding”) and think that software can be developed from first principles without any practical skills.

Conversely, we see people who are convinced that “good code” is everything and can be achieved with little more than a quick look at an online manual and a lot of cutting and pasting; …

My opinion is that both attitudes are far too extreme and lead to poorly structured, inefficient, and unmaintainable messes even when they do manage to produce minimally functioning code. ”

Bjarne Stroustrup (designer of the C++ programming language)
Paul Anderson
dcspaul@ed.ac.uk
http://www.inf.ed.ac.uk/teaching/courses/ijp
Recording ...

We are being recorded (hopefully) ....

But slides & sound only (no video)
Welcome to the IJP course (and perhaps to Scotland)

Beinn Tulaichean (a “Munro”) - August 2017
This Talk

Is about ... 
- The course philosophy
- Some common misunderstandings
- Some thoughts on writing good software
- Object-oriented programming
- The principles behind the first assignment

It is not about ... 
- Practicalities of the course
  - please read the website content carefully
  - pay attention to the things that you are asked to do
  - use the Piazza forum if you have any questions
- Details of the Java language (or libraries)
  - read the book
  - use the online resources
  - use the Piazza forum if you have any questions
Programming

Introduction to Practical Programming

IJP is not ...
- An academic software engineering course
- A course about the technical details of the Java language

It is about ...
- Creating real, working, practically useful applications
- Locating and making use of appropriate resources
- Writing “clear”, “correct” and “maintainable” code

• “The course taught programming from a fully practical perspective - no "Hello world" approaches to programming - a very different, and better, method of learning.”

• “I didn't have much programming experience before this course and no Java experience, so it was a great intro for me and I feel I accomplished and learned a lot. I also really enjoyed doing the assignments.”

• “It forced me to give more thought to object oriented design than I did before and also made me experience tasks I tended to skip in my amateurish software development past.”
Example Assignment

By Yarong Song
Why Java?

- “Object-Oriented”
- Commonly used
- Good IDE support
- Designed from the beginning to be object-oriented
- Cross platform: “write once, run anywhere”
- Wide range of supporting libraries
No Lectures

IJP has no lectures ...
- Nobody can “teach” you how to write real programs
  We can help you to learn - but it takes a lot of practice
- Lectures are not a good use of your time
- It is important that you learn to locate the materials yourself

But it it very well supported ...
- Labs, online Forum, carefully-designed exercises
- It is important to engage with these and manage your own time

- “Just consists of reading a book which basically can be done from home.”
- “It would be much better if more learning material on JavaFX are provided.”
- “The final assignment is really helpful. I actually realised a project totally by myself.”
Activities

The official allocation is

- 20 hours of lab sessions
- 40 hours of private study
  - watch the screencasts, read the book, do the exercises
- 40 hours on the two programming assignments
- Two lectures (including this one)

In addition to the labs ... 

- You will probably need to spend about one day per week
  - more if do not have much programming experience
  - more if you are aiming for a particularly good mark

You will need to manage your own time

- Aim to work steadily
- Beware of assuming that you can rely on previous experience
We attempt to provide a “realistic” assessment ...

- Based on writing real applications
  - no exams or unrealistic exercises
- Good code has to demonstrate a number of criteria
  - realistic “criteria” include “correctness”, “readability”, etc.
  - good marks require all of these to be good
  - no artificial “adding up” of marks
- You will be required to explain your design clearly, in English

We are **required** to mark to the “Common Marking Scheme”

- A good, complete solution will normally achieve about 70-80%
- Higher marks are very rare
- Other courses *should* follow this too!

• “Not allowing students to get more than 80 marks is unfair when compared to other courses. The highest attainable mark is 80 not 100..."
Typical Results

IJP 2016-2017

https://ease.groups.inf.ed.ac.uk/cgi/ijp/2016/cgi-private/graph.cg...
What Makes Good Software?

A nice interface?

Lots of “features”?

“Efficiency”?

By Nikos Katiortzis
What Makes Good Software?

- It “works!”
- Other people can read it and clearly see how it works
- We are confident that it works (and it will stay working!)
- It fits together with other people’s code so that we can build bigger things
- We (and other people) can maintain it, so that we can keep it working when circumstances change
Automated testing is a big help when writing real code

- If I change this line ...
  Will it break something somewhere in the other 100,000 lines?
- Java has a good testing framework (JUnit)
- It is easy to accumulate tests as the software grows

But it doesn’t “prove” that the software works

- Testing can only show that it doesn’t work
- And it isn’t always clear what conditions are worth testing
  - if it works for 2+3, will it work for 56734+9753?

You need to think about testing when you design the code

- Some things can be very hard to test if they have not been designed with testing in mind
Readability

Does this “work”? 

class Div{static $1 $_;class $1{void _$(String $_){System.//
out.print($_);}$1(){$(());}void _(){$int __,$,__$,$ $$,$__,a=(1<<5),
b=100,c=12,bc=a*c;b-=1<<4;while(bc>0){for($$=_$=__=______$=(int)
b;__$>($$-(1<<2));__$(""+(char)(__$)),__$-=1<<1)for(___-$=-9;>($
$-6);__-=1<<1,__$(""+(char)(__$!=b?_:a)));char S=$(char)(b+c+1)
;__$("te"+$S+(char)(c+(int)$S));bc--;}}}Div(){$=new $1();}public static void main(String []$){Div b=new Div();}}

We could run it and see ...

‣ But maybe it only works on Tuesdays?
‣ And what happens if we want to change it?
‣ Maybe it will delete all my files if I run it?
Comments & Code

Code is a way of communicating ideas between people

- As well as with machines
- Comments should work together with the code

```java
x = x * 1.15
// multiply x by 1.15
x = x * 1.15
price = price * 1.15
// tax rate only
// applicable in the UK
tax = 0.15 * price
price = price + tax
x = x * 1.2
// multiply x by 1.2
x = x * 1.15
```

Layout is important

This bug compromised the SSL encryption in all iPhones ... 

```
static OSS Status
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
         uint8_t *signature, UInt16 signatureLen)
{
    OSS Status err;
    ...
    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
    goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
    ...
fail:
SSLFreeBuffer(&signedHashes);
SSLFreeBuffer(&hashCtx);
return err;
}
```

http://www.wired.com/2014/02/gotofail/

Eclipse can help you (re)format your code
How “readable” is my code?

- You are probably as good a judge as someone more experienced!
- You don’t have an opportunity to compare your code with others
  - because it is assessed work

“Adaptive, comparative judgement”

- You get to see pairs of samples of work from other students
- You simply decide which of each pair you think is clearer

This means ...

- We can compare your opinions with the marker’s opinion
- You get to see other approaches (good and bad) to readability

This is a research project ...

- We hope that you will let us know what you think
  - there will some questionnaires ...
Good “large-scale” structure is essential

- Can different people work on different parts independently?
  - you won’t be able to write all the code yourself
- Can we change one part without affecting lots of others?
  - things will change!
- Can we isolate parts for testing individually?

An “object-oriented” approach is intended to help with this

- The code is structured into separate “classes”
- With clear “interfaces” which define their external behaviour
- Loose “coupling” is good ...
  - changing one class does not require lots of changes elsewhere
- High “cohesion” is good ...
  - classes have a single, well-focussed purpose

This is the most important thing to learn from this course

- You will need to demonstrate this in order to pass
An Example

Creag Meagaidh (from Paul's Service)
We may want to support different photo services
  - Flickr, Wikipedia, ...

We may want to provide alternative interfaces
  - Using buttons for selection
  - Using menus for selection

We may want to change the overall behaviour
  - Create a quiz program?

How can we arrange the code ..
  - So that different people can work on different parts
  - So that we can use existing code from elsewhere
  - So that we can test it easily and have confidence in the code
Composing Objects

Services
- Flickr
- Wikipedia
- PaulsService
- LocalService

Controllers
- SimpleController
- QuizController

Proxies
- CacheProxy
- RetryProxy
- RandomProxy
- SlowProxy
- UnreliablyProxy

Views
- ButtonView
- MenuView
BlueJ

http://www.bluej.org/
How do we test the controller?

- Without depending on the network & the interface & there user
Test Harness

No network required (may return different results each time!)
No interface required
No human interaction required
If you have not already ...
- Sign up for the lab sessions (now!)
- Register for the Piazza forum (now!)
- Catch up with the schedule

Look at the tips on the web site ...
- Think carefully about the structure of your classes & code
- Try to write clear, correct & testable code
- Make use of the Piazza forum and the lab sessions
- Exploit the class structure & the Java language
  - don’t just attempt to write C-style code in Java

Have fun!
Enjoy the Course!

http://www.inf.ed.ac.uk/teaching/courses/ijp