Comp30070: Object-Oriented Programming

Mel Ó Cinnéide,
Room B2.08,
CSI Building.

Lecture Times: Wednesday 3-4 and Friday 11-12.

Lab Times: Tues 9-11, Thurs 2-4
- You will be assigned a lab session (on SIS?).
- Labs started this week.

Teaching Assistant:
John Mark Swafford (johnmarksuave@gmail.com)
John Mark is your first port of call for any questions regarding the labs.

Module web page: csserver.ucd.ie/~meloc/30070/oop.htm

Module Description

This module reviews object-oriented concepts such as classes, objects, message passing, inheritance and polymorphism.

There is a strong emphasis on practical program development, for which **Ruby** is the language used.

We'll cover these areas:
- Introduction to Ruby
- Object-Oriented Programming using Ruby
- Advanced Object-Oriented Programming
- Advanced Ruby
- *Possibly* a section on C++ for comparative purposes

Marks Breakdown

There is a very heavy emphasis on practical programming in this module.

Marks breakdown:

Assignments:	20%
Written exam:	40%
Practical exam:	40%

More on the practical exam on a later slide.

Practical Work

One 2-hour lab session per week. You will either be working on a particular worksheet or doing an assignment.

There will be a number of programming assignments during the year. The deadline will be fixed and submission will be online.

These assignments are vital for both the final practical exam and the final written exam.

20% is awarded directly for the assignments. It is impossible to do well in the practical exam (40%) without having the experience gained from doing the assignments.

Programming experience cannot be crammed!

Contacting Me

You can email me at mel.ocinneide@ucd.ie but if you have a question that involves a bit of to- and fro-ing please call to see me.

You can call to see me any time, ask me a question during lecture, after lecture, or email me to arrange a time to meet.

If your question is related to the scheduling of practicals etc., please contact the Teaching Assistant, John Mark Swafford (johnmarksuave@gmail.com).

If your question is related to the programming assignments, e.g., a program bug you can't resolve, please try to get help from the tutors in the lab first.

Practical Exam

At the end of the semester there will be a closed practical exam held in the programming lab.

You will be given a programming assignment. It is an **open book exam**, but there will be no help available from tutors.

In order to achieve a good mark, it is essential that **your solution works correctly**. It should also be designed and written well.

I cannot emphasize strongly enough the importance of doing the assignments during the semester.

Practical Exam Date

The date and time of the practical exam has been set to Thursday evening in the last week of semester:

Thursday, December 1, 6pm to 9pm.

If you can't attend this exam, don't take this module!

If any other UCD module tries to schedule for this slot, tell the module coordinator you have an exam at that time.

Textbook

There's no recommended textbook for this module. Here are two good options:

- David Flanagan and Yukihiro Matsumoto, *The Ruby Programming Language*, O'Reilly, 2008. 4 copies in library.

- Dave Thomas, *Programming Ruby*, Pragmatic Programmers, 2009 (aka "the pickaxe book"). 4 copies in library.

You'll find lots of good material on the web.

Web Resources

There are many web pages devoted to Ruby. Here are some links:

www.ruby-lang.org
The main site for all things Ruby.

http://www.ruby-lang.org/en/documentation/quickstart/
has a useful "Ruby in Twenty Minutes" tutorial.

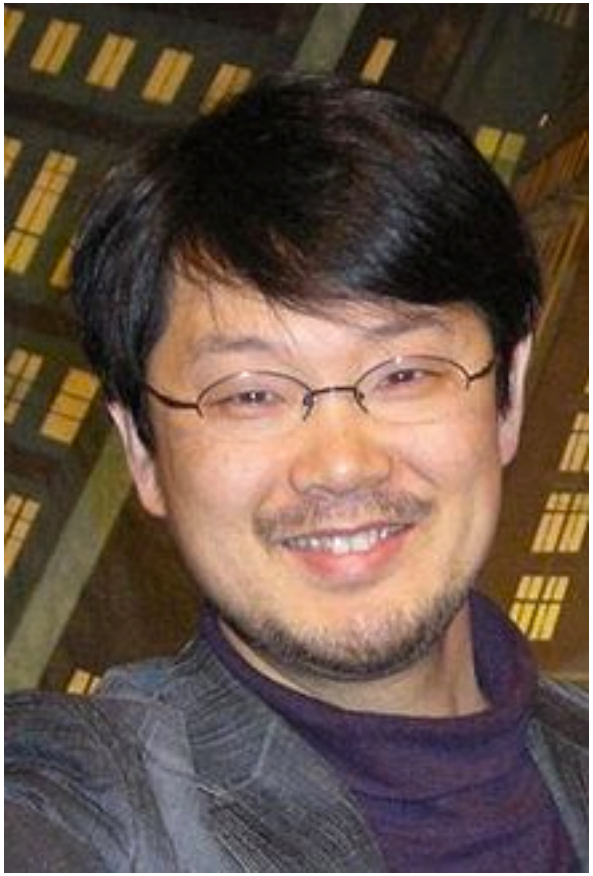
http://www.fincher.org/tips/Languages/Ruby
is a good Ruby tutorial. You may find one you prefer.

In the remainder of this lecture, I'll introduce **Ruby**.

Ruby Origins

Ruby was developed by a Japanese programmer, **Yukihiro Matsumoto** (aka "Matz" in Ruby circles).

His goal was to create an elegant language that would be easy to program in.



Ruby Origins

Ruby blends aspects of other programming languages like
- Perl,
- Smalltalk
- Lisp.

Ruby was released in 1995. Its popularity exploded with the release of **Ruby on Rails** in 2004.

(Ruby on Rails is a popular web application framework for developing web applications.)

Ruby is like Java!

- You have classes, objects, methods, inheritance.
- Method invocation uses the dot operator.
- Garbage collection
- Objects are strongly typed
- Methods can be **`public`**, **`private`** or **`protected`**.
- ...

We'll consider differences on the following slides.

Ruby **isn't** like Java!

- **Interpreted**, not compiled.
- All data fields are private by default
- Everything is an object, including numbers.
- No need for parentheses in method invocations
- Variable names have no type, and there's no static type checking. No need to use casting.
- **`public`**, **`private`** and **`protected`** have different meanings from Java
- Constructors are called **`initialize`**.
- ...

A simple Ruby program

Here we develop a **Greeter** program interactively...