

# CSC7322: Object Oriented Development

**J Paul Gibson, A207**

`paul.gibson@it-sudparis.eu`

<http://www-public.it-sudparis.eu/~gibson/Teaching/CSC7322/>

## **Introduction**

<http://www-public.it-sudparis.eu/~gibson/Teaching/CSC7322/L0-Introduction.pdf>

# From module description

## **Objectives :**

Expertise in developing object oriented models using a wide range of modelling languages and supporting tools,  
Ability to easily adopt new advances in object oriented development techniques.

**Keywords :** OO Software Development, OO Programming (Java), OO Modelling (UML)

## **Prerequisites :**

Some Programming Experience and Foundational Mathematics

# From module description

## Program :

The overall goal is for the students to understand why object oriented development has become the standard software engineering method of the 21st century. They will be able to explain the major variations in semantics between different OO modelling languages. They will master at least one OO Programming language (Java) and will be familiar with OO modelling languages during all stages of the development lifecycle (including UML). Finally, they will be able to apply OO techniques and tools in the development of simple components and services.

- From procedural programming to Object Oriented Programming : a history
- OO modelling : fundamental concepts – syntax and semantics
- OOP in Java
- OO Requirements Capture and Design with UML
- Object Oriented Formal Modelling – JML, UML2B, Objects as State Machines
- Objects as components
- Objects as services
- Distributed objects and inter-object communication
- Parallel Programming with objects
- Latest technologies and tools – aspects, ...

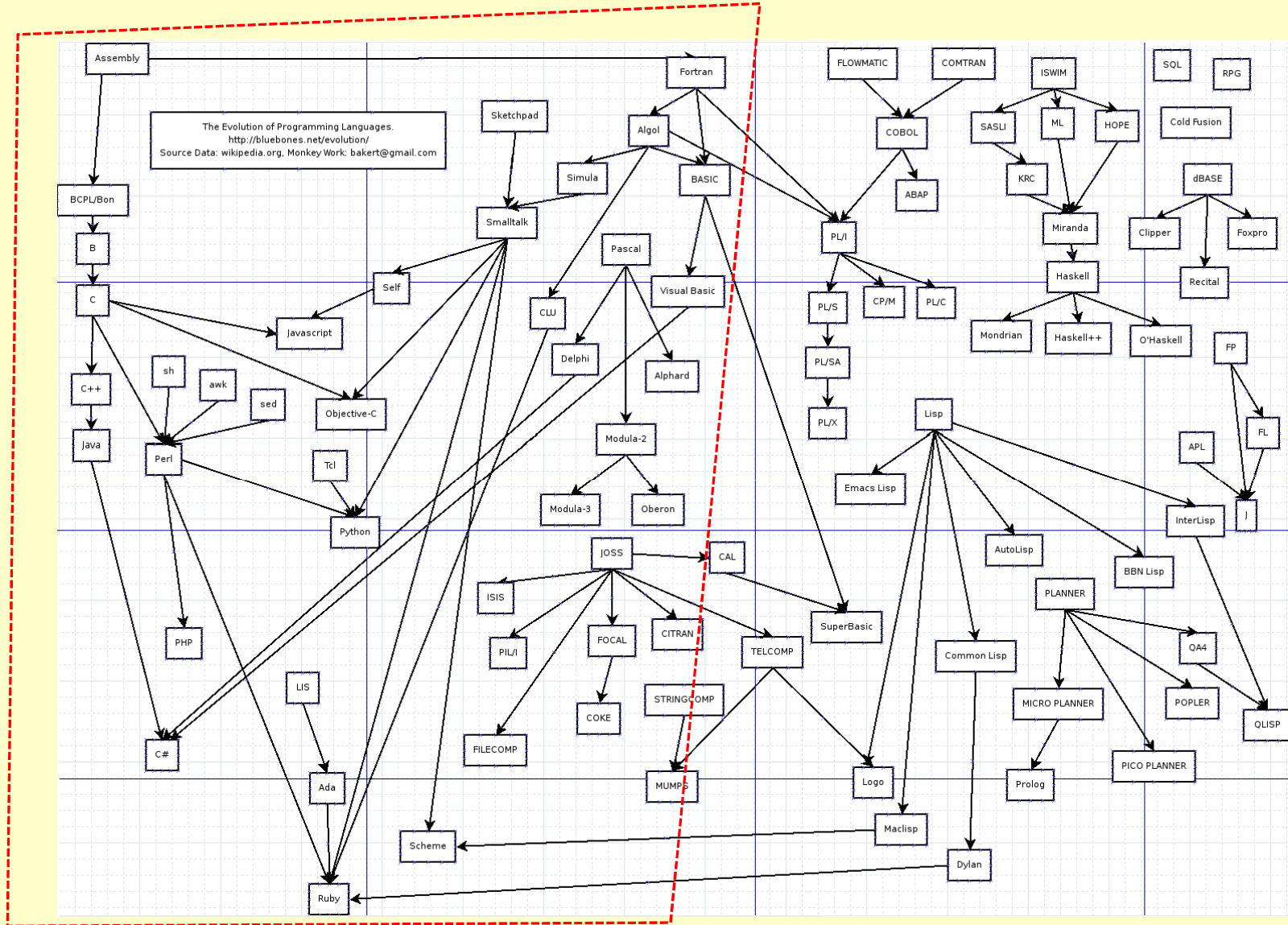
# Web Site:

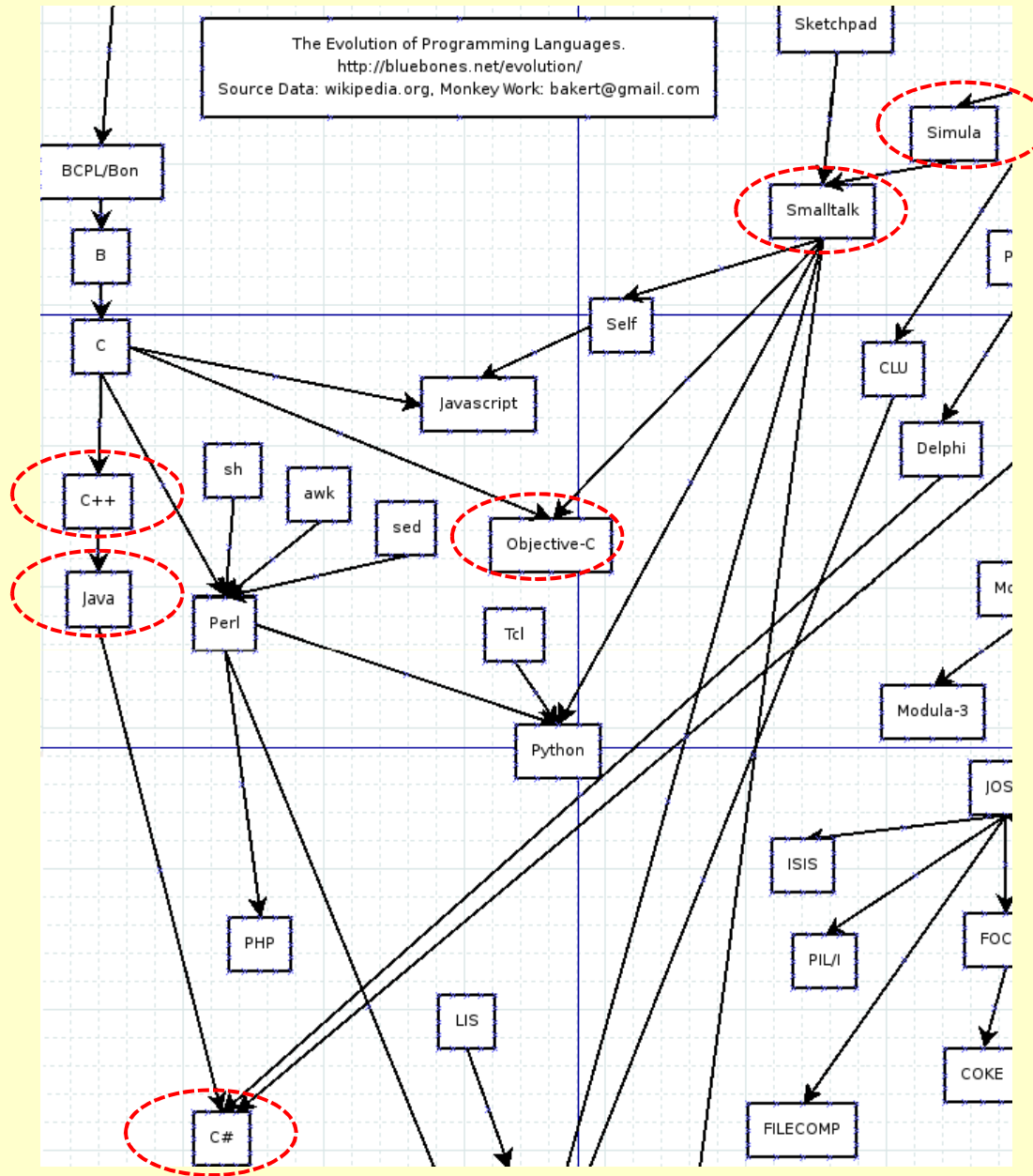
The screenshot shows a Mozilla Firefox browser window with the following details:

- Address Bar:** <http://www-public.int-evry.fr/~gibson/Teaching/CSC7322/>
- Page Title:** J. Paul Gibson's Teaching - CSC7322
- Page Content:**
  - Logos:** INSTITUT TELECOM and TELECOM SudParis.
  - Header:** Teaching-CSC7322 for Dr J. Paul Gibson, LORiciels-Réseaux (LOR), T&MSP SudParis, France.
  - Section: Object Oriented Development (OOD)**
    - Text: This module is a core part of the MSc programme *Software Engineering and Ambient Intelligence* ([Internal Moodle Link](#), [External TSP Link](#)).
    - Text: The material will be uploaded dynamically: the teaching approach is based on PBL and much of the learning will be through interaction/group work during the assigned lecturing time. (Please check the website for updates before every lecture.)
  - Section: Assessment**
    - Text: The assessment will be based on:
      - Individual Programming Practical Exam - 50% (details to follow)
      - Written Exam - 50% (details to follow)
  - Section: Sessions**
    - Text: Sessions are a mix of problem-based learning, group project work, directed practicals, interactive lectures and traditional lectures. There is no preset format - the lecturer organises the style and content of each session depending on the needs of the class.
    - Text: Please note that to access internet from C106 you may need to set the proxy server for your web browser to `proxy.int-evry.fr` and port 81.
  - Section: Session 1: Tuesday 24th January (13h45, C106) - Introduction**
    - Lecture Slides**
      - Text: Introduction: [pdf](#), [ppt](#)
    - Additional reading material**

The browser's status bar at the bottom shows a Google search URL: <http://www.google.fr/search?um=1&hl=en&client=firefox-a&rls=org.mozilla:en-GB:official&q=history+object+oriented+programming&ie=UTF-8&sa=N&tab=sw#q=history+objec...>

# History: OOP





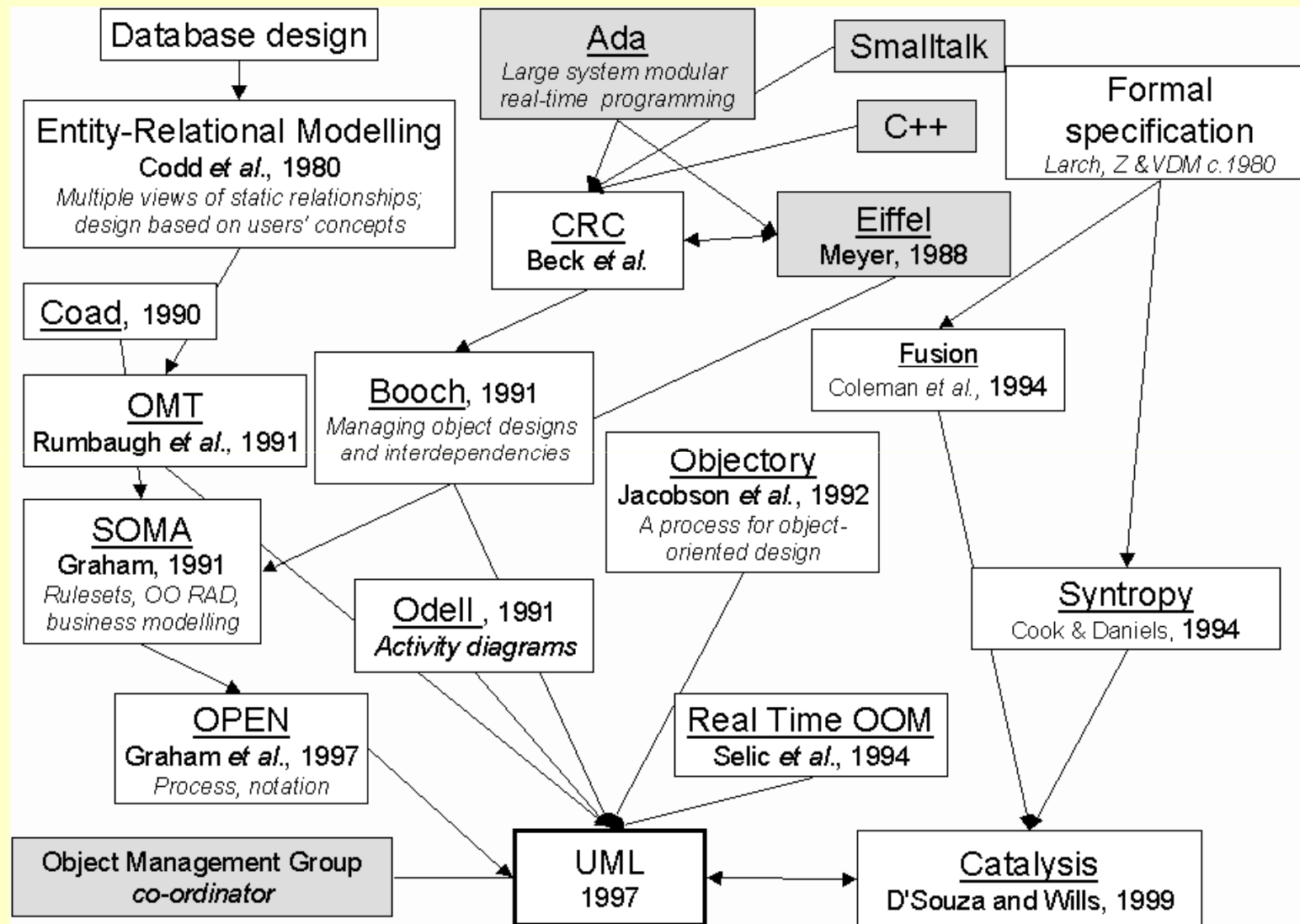
Languages  
 that we  
 will  
 discuss in  
 this  
 module

**Note:** each language has also evolved over time.


For example, **JAVA** has gone through 20+ updates since I started using it:

- 1 JDK 1.0 (January 23, 1996)
- 2 JDK 1.1 (February 19, 1997)
- 3 J2SE 1.2 (December 8, 1998)
- 4 J2SE 1.3 (May 8, 2000)
- 5 J2SE 1.4 (February 6, 2002)
- 6 J2SE 5.0 (September 30, 2004)
- 7 Java SE 6 (December 11, 2006)
  - 7.1 Java SE 6 Update 10
  - 7.2 Java SE 6 Update 11
  - 7.3 Java SE 6 Update 12
  - 7.4 Java SE 6 Update 14
  - 7.5 Java SE 6 Update 15
  - 7.6 Java SE 6 Update 16
  - 7.7 Java SE 6 Update 17
  - 7.8 Java SE 6 Update 18
  - 7.9 Java SE 6 Update 19
  - 7.10 Java SE 6 Update 20
  - 7.11 Java SE 6 Update 21
  - 7.12 Java SE 6 Update 22
  - 7.13 Java SE 6 Update 23
- 8 Java SE 7.0
- 9 Java SE 8.0

# HISTORY OOD (figure from [http://uml-tutorials.trireme.com/uml\\_tutorial\\_1.htm](http://uml-tutorials.trireme.com/uml_tutorial_1.htm))



## Some recommended reading ... with more to come in later lectures


*SIMULA: an ALGOL-based simulation language*, Ole-Johan Dahl and Kristen Nygaard, 1966  pdf

*Programming with Abstract Data Types*, Barbara Liskov and Stephen N. Zilles, 1974  pdf

*The Smalltalk-76 Programming System Design and Implementation*, Daniel H. H. Ingalls 1978  pdf


*Object-oriented programming: Themes and variations*, Mark Stefik and Daniel Bobrow, 1985  pdf

*Object-Oriented Development*, Grady Booch, 1986  pdf

*Genericity versus Inheritance*, Bertrand Meyer, 1986  pdf

*What is Object-Oriented Programming?*, Bjarne Stroustrup, 1988  pdf

*Virtual classes: a powerful mechanism in object-oriented programming*, Madsen, O. L. and Moller-Pedersen, B., 1989  pdf

*Applying "Design by Contract"*, Bertrand Meyer, 1992  pdf

*The Early History of Smalltalk*, Alan C. Kay, 1993  pdf

*Covariance and Contravariance: Conflict without a Cause*, Giuseppe Castagna, 1995  pdf

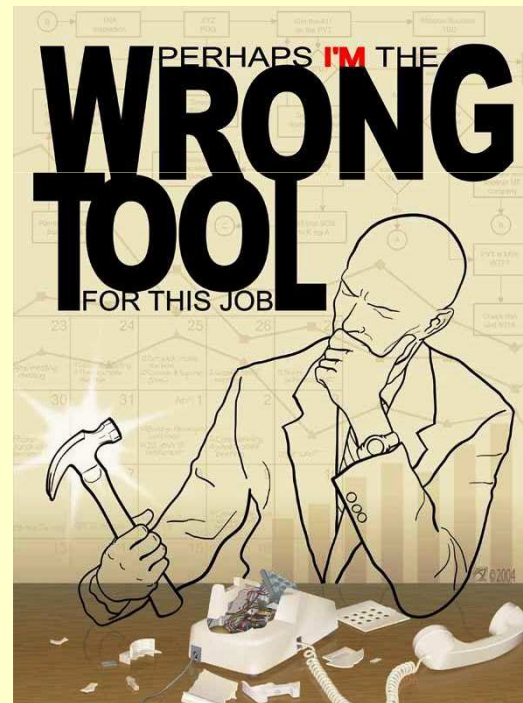
*A brief history of the object-oriented approach*, Luiz Fernando Capretz, 2003  pdf

# *Pseudo Object-Oriented Programming Considered Harmful*

Conrad Weisert  
Information Disciplines, Inc., Chicago

<http://www.idinews.com/westfall.pdf>

*If your only tool is a  
hammer, everything  
looks like a nail.*



## *Pseudo Object-Oriented Programming Considered Harmful*

### **QUESTION:**

Write a « hello world » program (or programs) that illustrate *your understanding* of 1 or more key object oriented concepts.

Present the program and the concept(s) to the class

A recent piece in *ACM Communications*<sup>1</sup> urged us to do more to indoctrinate our students into object-oriented programming. The author proposed to rewrite the popular "Hello, World" first complete program example "to include a user-created object". He believes this Java version to be object-oriented:

```
class HelloWorld {
    public static void printHello()
    {System.out.println
        ("Hello, world");
    }

    public class UseHello {
        public static void
            main(String args[])
        {HelloWorld myHello
            = new HelloWorld();
            myHello.printHello();
        }
    }
}
```