

CSC 7322 : Object Oriented Development

J Paul Gibson, A207

`paul.gibson@int-edu.eu`

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

Other Languages (OO)

[.../~gibson/Teaching/CSC7322/L6-OtherLanguages-PBL.pdf](http://www-public.it-sudparis.eu/~gibson/Teaching/CSC7322/L6-OtherLanguages-PBL.pdf)

PBL – programming in a *new* OO language

This lecture is PBL where you are to implement a solution to a problem in an OO language of your choice (other than Java).

To help you, I provide a solution in Java that you can choose to re-use in any way you wish (download [TrafficLightsJunction.zip](#) from the web site)

The languages you have chosen to work with are:

- OCaml
- C++
- Smalltalk
- C#

Introduction to OCaml

Recommended Online Reading

Using, Understanding, and Unraveling The OCaml Language: From Practice to Theory and vice versa, [Didier Rémy](#), 2002
<http://caml.inria.fr/pub/docs/u3-ocaml/index.html>

Introduction to Objective Caml, [Jason Hickey](#), 2008
<http://files.metaprl.org/doc/ocaml-book.pdf>

Other Online material

Download from: <http://caml.inria.fr/download.en.html>

Manual at: <http://caml.inria.fr/pub/docs/manual-ocaml/index.html>

Tutorial at: <http://www.soton.ac.uk/~fangohr/software/ocamltutorial/>

Introduction to C++

Recommended Online Reading

Thinking in C++, **Bruce Eckel**, 1995

http://www.codeguru.com/cpp/tic/tic_c.shtml

A tour of C++, **Bjarne Stroustrup**, 2004 (chp 3 of The C++ programming lang.)

http://www2.research.att.com/~bs/3rd_tour.pdf

Other Online material

Cygwin download (for Windows): <http://www.cygwin.com/>

DJGPP download (for Windows): <http://www.delorie.com/djgpp/>

Tutorial at: <http://www.cplusplus.com/files/tutorial.pdf>

Introduction to Smalltalk

Recommended Online Reading

Smalltalk Best Practice Patterns Volume 1: Coding, **Kent Beck**, 1997

<http://stephane.ducasse.free.fr/FreeBooks/BestSmalltalkPractices/Draft-Smalltalk%20Best%20Practice%20Patterns%20Kent%20Beck.pdf>

Early History of Smalltalk, **Alan C Kay**, 1993

<http://samizdat.cc/shelf/documents/2004/08.02-historyOfSmalltalk/historyOfSmalltalk.pdf>

Squeak by Example, **Black, Ducasse, Nierstrasz, Pollet**, 2009

<http://squeakbyexample.org//SBE.pdf>

Other Online material

Gnu download: <http://smalltalk.gnu.org/download>

Squeak smalltalk download: <http://www.c2.com/cgi/wiki?SqueakSmalltalk>

Introduction to C#

Recommended Online Reading

C# and the .NET Platform, **Andrew Troelsen**, 2003

http://content.imamu.edu.sa/Scholars/it/net/09978295_lese_1.pdf

Other Online material

C# Language Specification:

<http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf>

dotgnu project: <http://www.gnu.org/software/dotgnu/>

.NET Tutorial for Beginners:

<download.microsoft.com/.../dotnet%20tutorial%20for%20beginners.pdf>

PBL – Problem Rules

I will not help you with your programming – I want to see how each of you adapt to a *new* language/IDE.

If you get stuck then try to solve a simpler version of the problem that I give you

If you cannot manage to program anything useful for solving the problem set in your chosen language then change language or ask to work with another student.

You are to think about how useful the Java code has been to you in writing your own code.

If you solve the problem you are to prepare a short presentation of your solution to the other students: focus on the similarities/differences between Java and your chosen language.

The Traffic Light Junction Problem

At a road junction traffic lights control the flow of traffic in a safe fashion – so that accidents do not happen because 2 lights that control the same space go green at the same time!

In this problem we abstract away from the traffic light state to being a boolean: stop/red/False and go/green/True

Each junction has a number of lights each in a boolean state (>1, otherwise why do we need a controller?)

Each junction has a set of safety rules stating which lights can be green at the same time without a risk of an accident.

You are to code a junction system which has a single `changeLight` method which changes the state of the specified light in the junction without putting the junction in an unsafe state. In order to do so, it may – of course – need also to change the state of some/all of the other lights. How this is done is open to implementation freedom.

The Traffic Light Junction Problem: 2 solutions

- Simplest solution:
 - If the light we change goes green then turn all others red – advantage in its simplicity, but does not maximise traffic flow
- More complex solution which could improve flow:
 - If the light we change goes green then any light which is currently green (and should not be green at the same time as the newly changed light) must be turned red. The other green lights (that are allowed to be green at the same time as the newly changed light) do not need to change

TO DO: Code both these solutions in your language of choice

My Java solutions can be downloaded from the web site:

[TrafficLightsJunction.zip](#)

GOOD LUCK ☺