

## Programming Language Paradigms

### Assignment 2: Comparative Programming Languages

Write a report on **ONE** of the topics given below. Each topic presents a particular position in research on programming languages. Your report should be structured around this, i.e. composed of sections with suitable headings such as Introduction (stating the proposition and including background), Arguments For, Arguments Against, Conclusions, and, if necessary, further subsections. Include a bibliography (a couple of references are given for each topic to start you off). Also, where possible, include diagrams and simple example programs as this will make the report easier to understand.

References to articles should be given within your report in a form such as (Wilson and Clark, 2000) and a list of all references used must be given at the end of the report in the bibliography.

If you directly quote from an article, make this clear by using inverted commas or putting the quote in an indented paragraph on its own. You must also give a reference to the source of the quote.

Reports should be about 7-10 pages (in a reasonable font size). Very long reports will **not** be viewed favourably unless of exceptional quality. This assignment counts for **25%** of the final mark in 31Y4. It must be submitted in order for you to be given a grade for the course.

All articles and books listed below are available in the library in photocopy packages, as URLs or as reserved or popular loan books. Credit is given where reports give references other than those listed below. The given URLs should therefore be regarded as the starting point for an information search. These will be available from the 31Y4 home page, so you don't have to type in the URL!

Comparative language books are shelved in the library at SC 10 while texts on particular languages are at SC 10.6. The following journals, all of which are held in the library, may have useful additional references: Journal of Object-Oriented Programming; Software Practice and Experience; IEEE Software; ACM Sigplan. Introductions to most of the topics are given in Wilson and Clark.

**Date Due:**

**5pm Wednesday, April 20<sup>th</sup> April 2004**

Your report should be **printed out**, and **placed in the box labelled 31Y4** outside 4B89.

## 1. C# and Java

Microsoft's language C#, like Java, is derived from C++. The main difference is that Java was designed to be platform independent while C# is aimed specifically at programming in the Windows environment and is central to the .NET development.

Assess the importance of C#, how it satisfies its aims and to what extent it is superior to other C++ derivatives such as Java. Or is it just an attempt by Microsoft to dominate the market and force competitors out?

### References

The web has many articles on C#. The following give good starting points.

[www.javaworld.com/javaworld/jw-11-2000/jw-1122-csharp1.html](http://www.javaworld.com/javaworld/jw-11-2000/jw-1122-csharp1.html)

[www.javaworld.com/javaworld/jw-11-2000/jw-1122-letters.html](http://www.javaworld.com/javaworld/jw-11-2000/jw-1122-letters.html)

[msdn.microsoft.com/msdnmag/issues/0900/csharp/default.aspx](http://msdn.microsoft.com/msdnmag/issues/0900/csharp/default.aspx)

[www.softsteel.co.uk/tutorials/cSharp/cIndex.html](http://www.softsteel.co.uk/tutorials/cSharp/cIndex.html)

## 2. Visual Programming

Programs normally consist of ASCII text. Experimental languages have been developed in which the syntax consists of diagrams. Outline the main features of such languages and discuss their advantages and disadvantages for the programmer. Consider particularly the different approaches of a beginner programmer and a programmer of at least 5 years experience. Do you think visual languages might be better suited to one class of programmer more than another?

### References

K Kahn, *Drawings on Napkins, Video-Game Animation, and Other Ways to Program Computers*, Comm ACM, Vol 39, No 8, pp 49-59, August, 1996.

T Selker, *New Paradigms for Using Computers*, Comm ACM, Vol 39, No 8, pp 60-69, August, 1996.

D C Smith, A Cypher, J. Spohrer, *KIDSIM: Programming Agents Without a Programming Language*, Comm ACM, Vol 37, No 7, pp 55-67, July, 1994.

## 3. Built-in Verification Support

Some applications require a more rigorous approach to programming, with extra guarantees that the software behaves as expected (for example, safety critical systems). One method of doing this is to provide support for making assertions about the correctness of the code within the code itself, and building extra reasoning support into the compiler. Alternatively, one might use a standard programming language within a verification environment. Describe the pros and cons of each approach, with respect to verification in Java. A particular distinguishing feature is the willingness to hide the underlying mathematics from the programmer. Do you think this is a good thing?

## References

[www.eschertech.com/](http://www.eschertech.com/)

[portal.acm.org/citation.cfm?id=776816.776836&dl=portal&dl=ACM&type=series&idx=776816&part=Proceedings&WantType=Proceedings&title=International%20Conference%20on%20Software%20Engineering](http://portal.acm.org/citation.cfm?id=776816.776836&dl=portal&dl=ACM&type=series&idx=776816&part=Proceedings&WantType=Proceedings&title=International%20Conference%20on%20Software%20Engineering)

[sct.inf.ethz.ch/research/](http://sct.inf.ethz.ch/research/)

[www.dai-arc.polito.it/dai-arc/auto/tools/tool8.shtml](http://www.dai-arc.polito.it/dai-arc/auto/tools/tool8.shtml)

<http://www.praxis-cs.co.uk/sparkada/>

<http://www.cs.iastate.edu/~leavens/JML/>

<http://vl.fmnnet.info/b/>

## 4. Domain Specific languages

Most well-known programming languages are *general-purpose*, capable of being used in a variety of application areas. *Domain-specific languages (DSL)*, however, are designed to be used for specific kinds of applications. Giving some examples, outline the main features of domain-specific languages, comparing and contrasting them with a more general programming language, such as Java. While use of a domain specific language might make expression of a particular problem easier, do you think that it might limit you as a programmer to the “expected” solutions? That is, are you less likely to think outside the domain area, to find novel approaches to solving problems?

### References:

Call for papers for USENIX conference on DSLs (gives a nice overview of DSLs):

[www.usenix.org/publications/library/proceedings/dsl97/cfp.html](http://www.usenix.org/publications/library/proceedings/dsl97/cfp.html)

An overview of DSLs with lots of good links:

[compose.labri.u-bordeaux.fr/documentation/dsl/](http://compose.labri.u-bordeaux.fr/documentation/dsl/)

A paper describing a framework for DSLs:

[www.cs.unc.edu/~faith/DSLhtml/](http://www.cs.unc.edu/~faith/DSLhtml/)

An annotated bibliography:

[homepages.cwi.nl/~arie/papers/dslbib/](http://homepages.cwi.nl/~arie/papers/dslbib/)

## 5. Functional programming in the real world

The simple form of functional languages is both attractive to theoreticians and provides a good foundation on which to build and experiment with new ideas. However, with the exception of Lisp, functional languages are little used outside universities or research groups. What stops functional programming from being more widely used? Is it likely that more commercial applications for functional programming may be developed? Support your argument with facts about functional programming and a variety of computer applications.

### References

Runciman, C., Waleling, D. (1995). Applications of functional programming, UCL Press. (Available from library SC 9.1 RUN)

[homepages.inf.ed.ac.uk/wadler/realworld/](http://homepages.inf.ed.ac.uk/wadler/realworld/)

[www.math.luc.edu/icfp/](http://www.math.luc.edu/icfp/)

University of Stirling

Department of Computing Science and Mathematics

CES

[www-sal.cs.uiuc.edu/~kamin/cs497/](http://www-sal.cs.uiuc.edu/~kamin/cs497/)  
[haskell.cs.yale.edu/yale/papers/emsoft01/](http://haskell.cs.yale.edu/yale/papers/emsoft01/)

## 6. Combining Prolog and Java

It can be argued that as Prolog and Java have complementary strengths, a useful language could be created by combining them. Discuss the problems that users might have, given that the languages are so different, and other problems encountered (such as implementation problems). Weigh these against the benefits occurring because of the ability to use the most appropriate paradigm for a particular problem, and any other benefits which may occur to you.

### References

[www.ugosweb.com/jiprolog/index.shtml](http://www.ugosweb.com/jiprolog/index.shtml)  
[grunge.cs.tu-berlin.de/~tolk/vmlanguages.html#logic](http://grunge.cs.tu-berlin.de/~tolk/vmlanguages.html#logic)  
[www.amzi.com/products/prolog\\_products.htm](http://www.amzi.com/products/prolog_products.htm)

## 7. Python

Python is an object-oriented programming language that is used widely amongst the open source community and is gaining ground elsewhere. Like Perl, Python can be used as a scripting language used for to write Common Gateway Interface (CGI) scripts, but Python can also (like Perl) be used for other sorts of projects. The developers describe Python as an agile programming language, i.e. it is well suited for rapid application development.

Identify reasons for Python's popularity and the advantages of such a language over traditional languages such as C and Java.

### References

[www.python.org/](http://www.python.org/)  
in particular [www.python.org/doc/Comparisons.html](http://www.python.org/doc/Comparisons.html)  
[www.pythonology.org/home](http://www.pythonology.org/home)  
[diveintopython.org/](http://diveintopython.org/)

<p><b>You may also choose your own topic in Comparative Programming Languages provided you clear it with Dr Shankland before you start.</b></p>
---