

Principles of programming languages (2007)

Assignment

1. Aim The purpose of this assignment is to expose you to a novel programming language and to test your ability to evaluate and discuss about it using the major concepts studied in class.

2. Overview

You will work in teams. Your task is to select a less familiar programming language and evaluate its features. You can choose the language from the appended list or you can come with new proposals. You should register in the beginning of the course for a certain language by mailing me your choice (nsilvis@few.vu.nl). The first come-first served algorithm applies. Depending on how much people will attend the course we will have max 3 people working on the same language.

The report Your group must present its findings in a written report and deliver it on paper before December 1st 2007. In your report you should specify a brief history of the language including its inventor(s), its purpose and common uses and an evaluation of the language features. Describe the language features and what differentiates it from similar programming languages by using concepts learned during the PPL course, like name bindings and scope rules, types, call mechanisms, error handling, memory management, high order functions, polymorphism, object-orientation, etc. Show sample code fragments where relevant.

Further you have to perform a benchmarking experiment and compare your language performance versus Java. Your benchmark should implement the **Heapsort** program for an array of integers in both languages. The Java implementation of the **Heapsort** algorithm should be taken from Ref. [4]. You have to run both programs on the same hardware and register the elapsed CPU time or wall clock time required to execute the benchmark. Try to vary the array length (10, 100, 1000, etc) and register the effect on the speed. Analyze shortly your results. Mention in your report the execution environment used in each case.

You also have to include a bibliography of your sources.

Please note that delivering your written report in time is a condition to be admitted to the exam.

The presentation Each group will present its language (10 min) in class, during a session planned in week 48, on 30 November. During your presentation you have to show the unique characteristics and applications of your language and the benchmarking report. The idea is not to teach us a language but to open a discussion on different

programming languages, with the conclusion that there is no winner, but that each language is suitable for a certain purpose.

3. Grading

The final grade is calculated as $0.3 * \text{assignment} + 0.7 * \text{written exam}$.
The assignment is compulsory and will be graded as follows:

- Quality of the report: 75%
- Presentation: 25%

4. The list of languages to be evaluated

Fortran	Cobol	Scheme	Haskell
C#	Ruby	PHP	Perl
Javascript	Tcl/Tk	Self	Eiffel
R	Lisp	Postscript	Pascal

Other ideas are welcome.

5. References

1. <http://www.lib.uchicago.edu/keith/crisis/lang-features.html>, Features and misfeatures of Programming languages
2. <http://www.jvoegele.com/software/langcomp.html>, Programming language comparison
3. <http://www2.latech.edu/~acm/HelloWorld.shtml>, The ACM "Hello World" Project
4. <http://www.inf.fh-flensburg.de/lang/algorithmen/sortieren/heap/heapen.htm>, The heap sort algorithm, by H.W. Lang, F.H. Flensburg
5. J.M. Bull, L.A. Smith, L. Pottage and R. Freeman, Benchmarking Java against C and Fortran for scientific applications, Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande JGI'01, published in Concurrency and Computation: Practice and Experience, vol. 15, pg. 417, 2001
6. L. Prechelt, An Empirical comparison of seven programming languages, Computer, October 2000, p. 23.
7. <http://www.osnews.com/printer.php/5602/Nine-Language-Performance-Round-up-Benchmarking-Math-and-File-IO>. C. W. Cowell-Shah, Nine Language Performance Round-up: Benchmarking Math & File I/O, 2004
8. <http://www.idiom.com/~zilla/Computer/javaCbenchmark.html>, J.P. Lewis and U. Neumann, Performance of Java versus C++, 2004

APPENDIX. The Heapsort algorithm implemented in Java [from Ref.4].

The following Java class *HeapSorter* encapsulates the functions *downheap*, *buildheap* and *heapsort*. In order to sort an array *b*, Heapsort is called with the statement `HeapSorter.sort(b)`.

```
public class HeapSorter
{
    private static int[] a;
    private static int n;

    public static void sort(int[] a0)
    {
        a=a0;
        n=a.length;
        heapsort();
    }

    private static void heapsort()
    {
        buildheap();
        while (n>1)
        {
            n--;
            exchange (0, n);
            downheap (0);
        }
    }

    private static void buildheap()
    {
        for (int v=n/2-1; v>=0; v--)
            downheap (v);
    }

    private static void downheap(int v)
    {
        int w=2*v+1;    // first descendant of v
        while (w<n)
        {
            if (w+1<n)    // is there a second descendant?
                if (a[w+1]>a[w]) w++;
            // w is the descendant of v with maximum label

            if (a[v]>=a[w]) return; // v has heap property
            // otherwise
            exchange(v, w); // exchange labels of v and w
            v=w;           // continue
            w=2*v+1;
        }
    }
}
```

```
private static void exchange(int i, int j)
{
    int t=a[i];
    a[i]=a[j];
    a[j]=t;
}

} // end class HeapSorter
```

🏠 H.W. Lang, FH Flensburg (lang@fh-flensburg.de) © . Created: 25.02.1997 Updated:
06.02.2006 