Department of Computer Science
University of Pretoria

Programming Languages
COS 333

Practical Lab Experience 1: Research Assignment

July 26, 2016

1 Objectives

This practical lab experience aims to achieve the following general learning objectives:

- Provide experience in independent research, focusing on topics related to programming language theory;
- Give superficial exposure to some of the more esoteric topics related to programming languages, which are not the primary focus of the course, or the prescribed material;
- Provide some introductory experience in the use of the LaTeX typesetting system;
- Provide some introductory experience in the use of some special-purpose programming languages.

2 Plagiarism Policy

The Department of Computer Science considers plagiarism as a serious offence. Disciplinary action will be taken against students who commit plagiarism. Plagiarism includes copying someone else's work without consent, copying a friend's work (even with consent) and copying material (such as text or program code) from the Internet. Copying will not be tolerated in this course. For a formal definition of plagiarism, the student is referred to `http://www.ais.up.ac.za/plagiarism/index.htm` (from the main page of the University of Pretoria site, follow the *Library* quick link, and then click the *Plagiarism* link). If you have any form of question regarding this, please ask one of the lecturers, to avoid any misunderstanding. Also note that the OOP principle of code re-use does not mean that you should copy and adapt code to suit your solution.

3 Submission Instructions

The following submission requirements must be adhered to for this practical lab experience. Marks will be awarded for adherence to these guidelines (see Section 6):

3.1 Document

The final research report must be compiled using the LaTeX typesetting system. Documentation related to the LaTeX system is available from this practical's folder on the course site [5]. The Informatorium Linux installations include the commonly used teTeX implementation of LaTeX. The MikTeX system is available for free download, if you prefer to use Windows (see `http://www.miktex.org/`).

You must include a list of references for the sources you consult. All references must be cited at the appropriate location within each question. Because your references will be marked (see Section 6), ensure that there are sufficient (do not make unsubstantiated statements, unless they are clearly your own opinion) and that each is complete and correct. You may reference online sources. You will receive marks for managing your references with BibTeX. Documentation for BibTeX is also provided on the course site [6].

3.2 Upload

Upload your practical-related files as tar-gzip archives to the appropriate assignment upload slots on the course website. There is an upload slot for the research questions document, where you must upload a tar-gzip file (named `s99999999.questions.tgz`, where `99999999` is your student number), which must include your complete research report (both a compiled PDF file, and the complete LaTeX and BibTeX source). Each implementation task has a separate upload slot, where you must upload a tar-gzip file (named `s99999999.implX.tgz`, where `X` is the number of the implementation question, and `99999999` is your student number), which must include the complete source code implementation of the task. Multiple uploads are allowed, but only the last one will be marked. The deadline is **Monday, 8 August 2016, at 23:00**.

4 Research Questions [Total: 55]

Answer the following questions. Your answers should be as complete and clear as possible. Provide only information relevant to the question, and be as concise as possible. Overly verbose and lengthy answers will probably disadvantage you during the marking process:

1. **Explain** what the term "Turing complete" means, in terms of programming languages. [2]
2. **Explain** what an esoteric programming language (or *esolang*) is. [2]
3. The general consensus is that esoteric programming languages are little more than amusing diversions for computer science researchers. **Argue** for and against this viewpoint. [5]
4. Choose any two (2) esoteric programming languages. **Describe** each language in terms of its designer(s), year of initial design, general syntactic and semantic characteristics, and whether the language is Turing complete. For each language, provide a short example code snippet, to illustrate its general characteristics (you do not have to write the code yourself). [20]
5. Consider the Emacs text editor, which uses programming language integration in an interesting fashion. **Explain** how Emacs uses the integration of a programming language in its design, what this language is used for, and which programming language paradigm this programming language falls within. [3]
6. Consider the Scala programming language. **Identify** the programming language that Scala is based upon, and **explain** which main functionality Scala adds to this language. [2]
7. **Contrast** C++ and Objective-C in terms of which language each extends, what functionality these extensions add, and the philosophy employed by each in terms of how this extended functionality is implemented. Also **identify** a major application area within which Objective-C has been used. [5]
8. Consider the object-oriented Alice language. **Describe** the language in terms of its institution of origin, the general purpose of the language, the typical programming environment used to help achieve this purpose, and its general syntactic and semantic structure. [10]
9. Consider the concept of "Design by Contract" (DbC). **Explain** what DbC broadly entails. **List** two (2) languages that natively support DbC. [5]
10. **Explain** what the purpose of the Valgrind tool is. [1]

5 Implementation Questions [Total: 15]

Implement the following programs. All the required software is available under the Informatorium Linux installations. For each question, include at least one comment at the top of the program, that details the complete instructions for executing the program:

1. **Implement** a PostScript 3 program that draws a stick-man figure. An example of the desired output is given in a PDF file called `stickman.pdf`, on the course site. Some reference guides to the PostScript language are available on the course website [1][2]. [5]

2. LOGO is a functional programming language often used for turtle graphics [3]. UCBLogo is an open source implementation of the LOGO language. Documentation [4] on how to use UCBLogo is available on the COS 333 course page. The source code and a Linux implementation can be obtained from `https://www.cs.berkeley.edu/~bh/logo.html`. [5]

   Implement a Logo program that draws a set of circles arranged in a circle. It must be possible to specify the number of circles that have to be drawn in a simple fashion (i.e. with minor modification to the program source code).

3. **Implement** a gawk (GNU AWK) script that will process a rectangular matrix of values with any number of rows and columns, which is provided as an input file. The values in the input file should be assumed to be comma-separated. The script should print out the totals of the values in each row and column, as well as the total of all the values in the matrix. For example, assume that the following file is given as input to the script: [5]

   ```
   1,2,3,4,5
   6,7,8,9,10
   11,12,13,14,15
   ```

   The output produced by the script should then appear as follows:

   ```
   ========================
   Rectangular matrix totals
   ========================

   Input file:  input.txt

   Total for row 1:  15
   Total for row 2:  40
   Total for row 3:  65

   Total for column 1:  18
   Total for column 2:  21
   Total for column 3:  24
   Total for column 4:  27
   Total for column 5:  30

   Total for entire matrix:  120
   ```

   Your script should assume that rows with too few values are padded with zeros (in other words, if the first line in the example file were `1,2,3,4`, the script should assume that the row represents `1,2,3,4,0`). Test your script using a variety of different-sized matrices. Remember to include at least one comment that describes how the script should be executed. A reference guide for the AWK language is also provided on the course website [7].

6 Marking

The marks for this practical lab experience will be allocated as follows:

Category	**Mark Allocation**
Research questions	55 marks
Implementation questions	15 marks
Use of LaTeX and BibTeX	10 marks
Report structure	5 marks
References	10 marks
Language style, grammar and language	10 marks
TOTAL	**105 marks**

References

[1] Adobe Systems Incorporated. *PostScript® Language Reference*, third edition, February 1999.

[2] Adobe Systems Incorporated. *PostScript® Language Reference Supplement — Adobe PostScript 3 Version 3010 and 3011 Product Supplement*, 30 August 1999.

[3] Wikipedia, The Free Encyclopedia. Logo (programming language). Online: `http://en.wikipedia.org/wiki/Logo_(programming_language)`, accessed 24 July 2010.

[4] Brian Harvey. *Berkeley Logo 6.0 — Berkeley Logo User Manual*, 1993.

[5] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. The not so short introduction to LaTeX 2_ε, version 4.14, 4 April 2004.

[6] Oren Patashnik. BibTeXing, 8 February 1988.

[7] Arnold D. Robbins. GAWK: Effective AWK programming — a user's guide for GNU Awk, edition 4.1, April 2015.