Department of Computer Science
University of Pretoria

Programming Languages
COS 333

Practical 1: Research Assignment

February 16, 2026

---

1 Objectives

This practical 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 (including esoteric programming languages, domain-specific programming languages, and multi-paradigm programming languages);
- Provide some introductory experience in the use of the LaTeX typesetting system;
- Provide some introductory experience in the use of some domain-specific programming languages.

---

2 Plagiarism Policy

Plagiarism is a serious form of academic misconduct. It involves both appropriating someone else's work and passing it off as one's own work afterwards. Thus, you commit plagiarism when you present someone else's written or creative work (words, images, ideas, opinions, discoveries, artwork, music, recordings, computer-generated work, etc.) as your own. Note that using material produced in whole or part by an AI-based tool (such as ChatGPT) also constitutes plagiarism. Only hand in your own original work. Indicate precisely and accurately when you have used information provided by someone else. Referencing must be done in accordance with a recognised system. Indicate whether you have downloaded information from the Internet. For more details, visit the library's website: `http://www.library.up.ac.za/plagiarism/`.

---

3 Submission Instructions

The following submission requirements must be adhered to for this practical. Marks awarded for this practical require you to follow these guidelines exactly, and you may receive a zero mark for some or all of your submission if you fail to observe these requirements (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 `https://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. Note that this does not mean you must include a reference for every question (questions that ask for your opinion, for example, do not require a reference). You may use AI-based tools (e.g. ChatGPT or Gemini) only as a starting point for your research, but you may not copy AI responses verbatim. Also note that AI tools are prone to making mistakes, so do not accept a response without verification. You will receive marks for managing your references with BibTeX. Documentation for BibTeX is also provided on the course ClickUP page [6].

3.2 Upload

There is a separate upload slot for a PDF of your research questions report, the LaTeXsource for your research questions report, and each of the practical tasks. Upload your practical-related files to the appropriate assignment upload slots on the course ClickUP page. Multiple uploads are allowed, but only the last one will be assessed. The deadline is **Monday, 2 March 2026, at 12:00**.

- You are required to submit only your research questions PDF to a Turnitin assignment upload slot. Submissions will not be permanently uploaded to the Turnitin archive. Submission to Turnitin is required in order to receive a mark for your answers to the research questions. Turnitin also performs AI checking, and you will be penalised if it is found that you've copied AI responses verbatim.
- You must also upload the LaTeX and BibTeX source for the research questions report in a single zip archive. You must include a compiled PDF file, and the complete LaTeX and BibTeX source.
- For your R and UCBLogo implementations, submit only the source files (that is, not a compressed archive). For your Inform 7 implementation, submit a zip archive of your project directory. It must be possible to run your submissions with only the files you upload.

The reports will be assessed offline by the tutors. The implementation questions will be assessed during the practical sessions in the week of **Monday 2 March 2026**. If you do not demonstrate your practical implementations during one of these sessions, you will receive zero for those questions.

4 Research Questions [Total: 20]

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 an esoteric programming language (or *esolang*) is. [1]
2. Esoteric programming languages are usually considered to have no real-world applications, and are therefore usually considered to be useless. **Suggest** four useful contributions that esoteric programming languages can make. [4]
3. Choose any two esoteric programming languages. **Describe** the languages in terms of their designer(s), year of initial design, and their general syntactic and semantic characteristics. Provide a short example code snippet, to illustrate the language's general characteristics (you do not have to write the code yourself). [5]
4. Consider the programming language SALT (Structured Assertion Language for Temporal Logic). **Briefly explain** where SALT was developed, what type of programming language SALT is, and what advantage SALT provides for a programmer. Also **suggest** two real-world programming domains that would greatly benefit from the use of SALT. [5]
5. About a year ago Andrej Karpathy, a founding member of OpenAI and the former director of AI at Tesla, posted about what he calls "vibe coding". See `https://x.com/karpathy/status/1886192184808149383` for more details. **Briefly explain** what "vibe coding" entails. and **briefly describe** two potential advantages and two potential drawbacks to this practice. [5]

5 Implementation Questions [Total: 10]

Implement each of the following programs. For each question, include at least one comment at the top of the program, that details the complete set of steps for executing the program:

5.1 UCBLogo [5]

Logo [2] is a dialect of Lisp (this means that Logo is at its core a functional programming language like Lisp). Logo is a domain-specific programming language that was primarily intended as a school-level educational programming language. Logo is most often used for turtle graphics, in which drawing and movement commands are used to create vector graphics.

UCBLogo [3] is an open source implementation of the Logo programming language developed at the University of California, Berkeley. Documentation [4] for UCBLogo is available on the COS 333 ClickUP page. Implementations of UCBLogo can be obtained from `https://people.eecs.berkeley.edu/~bh/logo.html`.

Implement a UCBLogo program containing a recursive procedure that draws a regular polygon with any specified number of sides. A regular polygon is equiangular (i.e. all vertex angles are equal) and equilateral (i.e. all sides have the same length).

It must be possible to specify the number of sides the polygon should have by means of a procedure input (you will know procedure inputs as function parameters). Note that the preferred mechanism for repetition in Logo is recursion, and therefore your procedure must be recursive. You will forfeit marks if you use an iterative structure such as `REPEAT`, `WHILE`, or `UNTIL`

Upload only the source code for your UCBLogo program to the appropriate upload slot on the course ClickUP page. Name your file `u99999999.lg`, where `99999999` is your student number.

5.2 gawk [5]

AWK [1] is a domain-specific programming language, intended for text processing and data extraction. The gawk (GNU AWK) release is an open source implementation of the AWK programming language. A reference guide for the AWK language [7] is provided on the course ClickUP page.

Implement a gawk 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. An example input file, named `input.txt`, is provided on the COS 333 ClickUP course page. 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:

```
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.

Upload only the source code for your gawk program to the appropriate upload slot on the course ClickUP page. Name your file `u99999999.awk`, where `99999999` is your student number.

6 Marking

The marks for this practical will be allocated as follows:

Category	**Mark Allocation**
Research questions	20 marks
References	5 marks
Use of LaTeX and BibTeX	5 marks
Implementation questions	10 marks
TOTAL	**40 marks**

References

[1] Wikipedia, The Free Encyclopedia. AWK. Online: `https://en.wikipedia.org/wiki/AWK`, accessed 16 February 2026.

[2] Wikipedia, The Free Encyclopedia. Logo (programming language). Online: `http://en.wikipedia.org/wiki/Logo_(programming_language)`, accessed 16 February 2026.

[3] Wikipedia, The Free Encyclopedia. UCBLogo. Online: `https://en.wikipedia.org/wiki/UCBLogo`, accessed 16 February 2026.

[4] Brian Harvey. *Berkeley Logo 6.2: 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 6.2, 28 February 2018.

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

[7] Arnold D. Robbins. GAWK: Effective AWK programming — a user's guide for GNU Awk, edition 5.3, 2024.