

COS 301 Programming Languages Technical Writing

Acknowledgment

- These notes are partially based on slides prepared by Josh Murphy, PhD candidate at the University of Maine, presented to students in COS 301 Programming Languages Fall 2007 and Fall 2008
- Some examples used herein are used with permission from some of my students in Fall 2008

Audience

- Whenever you are writing, you are writing “for” someone – the audience
- The tone and style of your writing should vary with the audience. Imagine writing about programming for:
 - Elementary school students
 - High school students
 - Your peers in college
 - A newspaper
 - A PhD committee
 - A trade journal
 - An academic journal

Audience for COS 301

- One of the major audience considerations is how much background knowledge can be assumed and how much must be explained
- Consider your audience for COS 301 to be your peers
- You can assume that they know about things like variables, objects, classes and types
 - They may not know about things like array slices, monads, exception handlers etc.

Readability

- You should strive to produce prose that is readable and informative
 - The reader should not have to read a paragraph or section twice to grasp its meaning
 - The reader should be able to follow your train of thought and understand the concepts that you are trying to convey
- In measuring readability, the two main metrics are sentence length and word complexity (number of syllables)
 - Words containing more than two syllables are considered “hard”
 - Long sentences are more difficult to read than short sentences

Readability

- You may have noticed that some academic journals are rather difficult to read. Why?
- Your goal in writing for this class is not to show off your erudition or impress me with your knowledge – it is to produce clear, concise and easy-to-read prose
- If you are familiar with reading grade level, you can consider grade 12 to be a target level

Tone and Style

- One of the goals of this course is to develop some skills in written presentation of technical material
- You will be writing within a highly-structured formal framework but your writing does not have to be stiff, stilted and overly formal
- In each paper that you write, consider that there are two goals:
 1. To convey to the reader an overview of the language aspect or aspects covered in the paper
 2. To convey to the reader your personal assessment of language features

Tone and Style

- In past classes I have noted that when students strive to be too formal, the writing style suffers greatly
 - However, the same students often write well when they feel less constrained – for example, in a take-home exam essay that asks only for their opinion
- Your goal is to communicate with your peers.
 - You are not writing a textbook for beginning programmers
 - You are not writing to impress me with your esoteric knowledge

So ...

- Write naturally, but avoid excessive informality
- Do not write dense verbiage that conveys what should be obvious to your peers
- Feel free to express an opinion, but support it:
 - With arguments
 - With citations (opinions of others)
 - With examples

Avoid excessive informality

- Avoid colloquial language, unless it is used to make a point

"Are Java and Javascript programming paradigms similar? Sort of ... maybe."
"As you can see, there were a lot of scripting languages floating around (and more today)"
"On top of the numeric data types, Python also has native container types and strings"
- Use the first person ("I ...") judiciously, and use the third person ("You ...") even more judiciously
 - "You can call things such as the square root function by first importing sqrt from math ..."

Don't state the obvious

- Remember your audience
 - "When a program is running, regardless of what language it is written in, it is allocated a certain amount of memory to perform its tasks."
 - "You can use logical NOT to say something like, $x \neq 2$, which will evaluate to true if x is equal to anything but 2, and false if x is equal to 2."
 - "Python's syntactic structure is similar to Java and other curly brace languages, but different because of the lack of curly braces."

Do express opinions (and support them)

- Perl is a unique, interesting, and capable language that has been quite popular since its introduction. The language is best suited for web programming, file and text manipulation, and common UNIX system tasks. It allows tasks that used to take multiple programming languages and hours of time to be completed in mere minutes, which gives credence to its nickname, "The Swiss Army chainsaw of programming languages". It is one of the most flexible languages ever designed, which reinforces its other slogan, "There's more than one way to do it". All of the flexibility, power, and expressiveness of the language can either be viewed as an advantage or disadvantage for Perl. Very experienced Perl programmers will argue that it is an advantage that no other languages have. Proponents of other languages argue that the language is too big. The C language, for example, is very small. The C Programming Language (Kernigan, Ritchie), the de facto reference book on C that covers the entire language is only 294 pages. The de facto reference that covers the entirety of Perl, Programming Perl (Wall, Christiansen, Orwant) is 1092 pages.

The Mayfield Handbook

- The Mayfield Handbook of Scientific and Technical Writing
 - An excellent resource and reference for all types of technical writing
- Abbreviated version is available online at <http://www.mhhe.com/mayfieldpub/tsw/toc.htm>

Basic Elements of Writing

- If you have mastered these, you're 90% there...

 1. Spelling
 2. Word Usage
 3. Grammar
 4. Sentence Structure
 5. Paragraph Structure

Spelling

- The most fundamental element of writing – regardless of what else is wrong, spelling should be correct
- If you are not sure how to spell something, use tools:
 - Spelling checker (don't rely on it blindly!)
 - Dictionary
 - Thesaurus
 - Google
- It also helps to know how to pronounce words
- Try [howjsay.com](http://www.howjsay.com)
<http://www.howjsay.com/index.php>

Word Usage

- Dictionaries and thesauri are helpful in describing how to use a word
- It's a good idea to have a dictionary handy whenever you write, especially if you are unsure of spelling and usage
- English is rich in homonyms or near-homonyms that are commonly misused. Ex:
 - complement, compliment
 - affect, effect
 - accept, except

Word Usage – Formal Style

- Reports for this class are to be written in a formal style
- The guidelines below are not absolute. Skilled writers may make exceptions
 - Avoid first person constructions
 - Avoid contractions
 - Avoid colloquial language
- The notation “informal” on your papers means that a word, phrase or sentence is too informal

Grammar

- Grammar is the set of rules by which sentences are constructed
- See Mayfield handbook chs 11-14 for a basic review
- Poorly constructed sentences may still be grammatically correct

Sentence Structure

- Probably the most common type of problem in papers
- Mayfield Ch. 6 classifies common errors

Stacked Modifiers and Nouns
Wordiness
Overloaded Sentences
Sentence Fragments
Comma Splice
Fused Sentences
Stringy Sentences
Agreement
Lack of Parallelism
Choppy Sentences
Misplaced Modifiers
Dangling Modifiers
Double Negatives
Faulty Comparisons
Inappropriate Shifts
Sequence of tenses
Pronoun Reference
Pronoun Case

Sentence Structure

- It's worth the few minutes it takes to review these types of errors
- Sometimes we will note a specific type of sentence error when grading your papers, e.g., "fragment" or "comma splice."
- Often a sentence will be noted simply with "rewrite" or "awk" (awkward).

Paragraph Structure

- Paragraphs consist of several sentences in a group. Together the sentences discuss one subject, often introduced in the first sentence (sometimes called the *topic sentence*)
- The topic sentence is usually the first sentence in the paragraph and is often the most general sentence
- All other sentences should contribute to an understanding of the topic sentence
- Paragraphs should follow a logical organization with a beginning, middle and end

Coherence

- Coherence definitions:
 - "the quality of being logical and consistent"
 - "capable of thinking and expressing yourself in a clear and consistent manner"
 - "logical and orderly and consistent relation of parts"
- Coherence in writing
 - The writing flows clearly, logically and is consistently organized
 - Each paragraph contributes to the meaning of the paper
 - Each sentence contributes to the meaning of the paragraph
 - Each phrase contributes to the meaning of the sentence

Cohesion

- "The big parts of a story should stick together, but the small parts need some stickum as well. When the big parts fit, we call that good feeling coherence; when sentences connect, we call it cohesion." [1]
 - (Roy Peter Clark, *Writing Tools: 50 Essential Strategies for Every Writer*. Little, Brown, 2006)

Paragraph Types

- Paragraphs follow typical patterns based on the mode or purpose of the paragraph:
 - Exemplification
 - Narration
 - Process
 - Description
 - Comparison and contrast
 - Analogy
 - Cause and effect
 - Classification and division
 - Definition
 - Analysis
 - Enumeration
- Let's look online at some descriptions

Types of Technical Writing

- There are many different types of technical writing, such as proposals, literature reviews, theses, technical manuals, end-user manuals, journal articles, etc.
- See Mayfield Ch. 2
- Each type of technical writing follows a specific format, with some variation
- We are primarily interested in Reports.

Reports

- Reports can be further categorized:
 - Laboratory Reports
 - Research Reports
 - Research Articles
 - Design and Feasibility Reports
 - Progress Reports
 - ...

Research Reports

- From the Mayfield Handbook

“Research reports present the results of formal investigations into the properties, behavior, structures, and principles of material and conceptual entities.”

The Research Report Process

1. Define the problem
 2. Research the problem
 3. Present your research in a focused, coherent document
- Let's talk about exactly what the “problem” is for COS 301
 - Properties, behavior, structure etc of a language
 - Design principles embodied in the language

Research and Survey Reports

- A research report may have quite a few sections, following a defined sequence:
 - 3.4.1 Introduction
 - * 3.4.1.1 Problem Statement
 - * 3.4.1.2 Purpose
 - * 3.4.1.3 Scope
 - * 3.4.1.4 Authorization
 - 3.4.2 Background
 - 3.4.3 Theory
 - 3.4.4 Design and Decision Criteria
 - 3.4.5 Materials and Apparatus
 - 3.4.6 Procedure
 - 3.4.7 Work Plan
 - 3.4.8 Results
 - 3.4.9 Discussion
 - 3.4.10 Conclusion
- In this class the “research” is more like a survey of existing knowledge

Survey Reports

- In this class your reports are a survey of existing knowledge rather than creation of new knowledge by research.
- For each report you will be given a list of topics
- Your goals are to clearly and concisely present material about each topic (not easy!) for your chosen language.

“I am sorry to have wearied you with so long a letter but I did not have time to write you a short one” (attributed to Blaise Pascal)

Report Structure

- Research reports are divided into front matter, body and end matter
- Front matter:
 - Title page
 - Abstract
 - Table of Contents
 - List of Figures
- Front matter is numbered with lowercase Roman numerals.
 - The title page is page i but is not usually explicitly numbered. The abstract is page ii, etc.

Sections and Subsections

- Reports should be organized into sections and subsections
 - Each section is distinct part of the report and has a descriptive heading
 - Subsections are distinct parts of the section and have descriptive headings
 - A subsection may be as short as a single paragraph but typically contains two or more
- The collection of sections and subsections is an outline of the document

Sections and Subsections

- Organization into sections and subsections helps the reader understand the report and contributes to readability
- It is often helpful to outline the report before starting to write – it helps you to organize your thoughts
- Number your sections and subsections, e.g.
 - 1. Data Types
 - 1.1 Scalar Types
 - 1.2 Arrays and Structures
 - 1.2 Objects
- You may use one or more additional levels if desired

Title Page

The title should say something meaningful



Abstract

- The abstract is a brief summary consisting of a single paragraph, several sentences and typically less than 150 words
- It provides a concise summary of the problem, method, results and conclusion of your report
- For this class method and results may not apply
- Always write an abstract

Abstract

Abstract

This paper is a survey of typing, scoping, and binding in C++ and Python. It surveys both language-supplied and user-defined data types, how these data types can be extended, and how scope and binding are handled in each language. These features are compared and contrasted in the two languages and the paper concludes with a summary of strong points and weak points in each language.

Table of Contents

- Most word processors can generate these automatically. Review your word processor's help files and learn how to do one.
- Do include front matter, except the TOC itself; body and end matter.

Table of Contents

Table of Contents	
Abstract	1
List of Figures	iii
1. Data Typing C++ and Python	1
2. Data Types in C++	2
2.1 Basic Data Types	3
2.2 Complex Data Types	4
3. Data Types in Python	6
3.1 Basic Data Types	6
3.2 Complex Data Types	7
4. Libraries	8
4.1 Libraries in C++	8
4.2 Libraries in Python	9
5. Scope and Binding	9
5.1 Scope and Binding in C++	8
5.2 Scope and Binding in Python	10
6. Comparisons and Conclusions	11
References	12
Appendix A C++ Code	14
Appendix B Python Code	19

List of Figures

- If you have figures (including tables) in your text they are indexed separately in a List of Figures
- It may be convenient to copy tables, lists and other illustrations from web references when covering language features such as operators or types
 - There is no reason not to do this as long as you cite your source!
 - These would typically appear in a list of figures
 - Note that large tables and figures copied from sources are not considered in the written page count for assignments

List of Figures

List of Figures	
Figure 1 C++ Data Types	3
Figure 2 Python Data Types	6
Figure 3 Dynamic Scoping in Python	9

Figures

- In addition to graphical illustrations, figures can include tables and even programming code examples.

That said, JavaScript also supports the *foreach* loop—an overload of the *for* construct. It is aware of JavaScript's array type and is capable of generating iteration over it (although, as a mostly-typeless language, some care must be taken that the loop fits the general case for all expected types that may be contained in the array). Figure 9 contains an example.

```
var arrayList = [ element1, element2, element3 ];
for (var countVar in arrayList)
{
  // do something
}
```

Figure 9. JavaScript's *foreach* equivalent.
In virtually all other ways, JavaScript parallels the behavior of C.

The Report Body

- Starts with the title (yes again) at the top of the page
- Has an introduction
- Presents research or survey information
- Presents conclusions
- Usually your report body will reflect the topics assigned in the project

The Basic Rule of Reporting and Presenting

1. Tell them what you're going to tell them (introduction)
 2. Tell them
 3. Tell them what you've told them (summary and conclusions)
- Typically in an oral presentation brief conclusions may be presented at the beginning
 - In a written research or survey report brief conclusions are presented in the abstract, but not the report introduction

Support and Supporting Material

- When writing about a new language you will often be expressing your opinion about the language
 - Opinions are fine things but only worthwhile if backed up with facts, citations and/or arguments
 - In discussing a programming language, facts may be hard to come by to support an opinion such as "this programming language is not well suited to scientific computing."
 - But you can support your opinions with citations or arguments
- Adequate support contributes to coherence

Examples of Support

- " All though Perl supports all of the above syntax for subroutines, the most common by far is sub NAME BLOCK. As will be shown in section 2.3, all though compile-time argument checking (PROTO above) is allowed in a limited form, arguments are typically accessed through the special variable @_. The support for prototypes and attributes (ATTR above) is limited and is only used in very specific circumstances [21]. As such, it will not be detailed in this report."
- " Perl attempts to meet every single need of its users, but I believe that no language is capable of that, and that Perl falls very short of that goal. It appears that over time features have been added to the language just because some other language has supported them. The object-oriented features added in Perl 5 feel like a half-hearted attempt and their usage can be quite dangerous since private instance variables do not exist. It is reminiscent of attempting object-oriented programming in C, which only contains structures."

Introductions

- Reports are typically divided into numbered sections
- The report itself, and each major section, begins with an introduction
- The introduction is not labeled as such
- This is a lot like paragraph structure (topic sentence, supporting sentences, conclusion) on a larger scale
- The introduction provides motivation for a report or section and identifies topics, problems, or issues
 - It does not present a conclusion. It is not an abstract!

Example

Control Structures in Simula and JavaScript: A Comparison

I. Control Structures

At its most basic, almost all tasks in the real world are a linear set of steps. Even the most complex tasks, be they in academia, the workplace, or the home (anyone who might question this list has obviously never attempted to assemble a modern home theater system), can be broken down into a set of discrete steps or sub-tasks that must be accomplished in order—there may be some variation of independent parts, but the overall "direction" of the task as a whole remains the same. There are times when these sub-tasks are themselves recursively broken down into further smaller tasks, but that goes further than the analogy used herein really should.

Where these real-world tasks draw a very close parallel to those of computer programming—the topic of this paper—is in the area of control flow. A human performing some real-world task must make decisions. The direction given a person as part of some task might be "for every object in this collection—say, for every soup can in the box—perform a task—in this case, checking the can's expiration dates, then put all the objects that meet a given criteria—here, having not expired yet—and perform an action with those objects—in this example, placing them on the shelf." These directions may be as layered and complex as the individual human's capability of understanding them, with potentially involved decisions and, sometimes, repetitions (in the above example, in the form of "for every object in this collection") throughout the task's process. However, an overriding feature of these sorts of tasks is that the steps are linear. They occur one at a time and while steps might be repeated, the repetition is still singularly linear.

Surveying a topic for a language

- It is very easy to fall into "laundry list" mode where you simply provide a "laundry list" related to the topic in question
- For example, when describing types available in a language, it is tempting to simply list the types with a brief description

Remember your audience

- This is where it helps to remember your audience
 - You can assume that your audience already knows about the basic scalar types (characters, integers, floats, booleans) available in most languages
 - Instead of a laundry list, you might present a table of types, and indicate that a language supports the usual scalar types and then go on to describe more interesting points of the type system

Including Tables and Lists

- Short tables and lists (typically 1/3 of a page or less) may be included in the text
- Longer tables or lists should be included as appendices

Reports and Programs

- Each assignment has an associated program that you will write, usually related to the topics of the report
- You can include a section that discusses how the program was written, how it operates, and how it demonstrates features of the language
- The program itself will go into an appendix

Typography

- Whether including program code in the body of the report or in an appendix, use a monospaced font such as Courier New or Lucida Console
- Compare this code

```
function showdiv(id) {  
  //safe function to show an element with a specified id  
  
  if (document.getElementById) { // DOM3 = IE5, NS6  
    document.getElementById(id).style.display = 'block';  
  } else {  
    if (document.layers) { // Netscape 4  
      document.id.display = 'block';  
    } else { // IE 4  
      document.all.id.style.display = 'block';  
    }  
  }  
}
```

Typography

- With this code

```
function showdiv(id) {  
  //safe function to show an element with a specified id  
  
  if (document.getElementById) { // DOM3 = IE5, NS6  
    document.getElementById(id).style.display = 'block';  
  } else {  
    if (document.layers) { // Netscape 4  
      document.id.display = 'block';  
    } else { // IE 4  
      document.all.id.style.display = 'block';  
    }  
  }  
}
```

End Matter

- End matter typically includes references, appendices and indexes
- For this class you have references and appendices (your program and output)

References

- In publications and proposals one of the cardinal rules is that any reference listed in a reference section or bibliography must be cited in the text
- For this class we will ignore this rule. You will develop and maintain an annotated bibliography throughout the semester
 - References cited in earlier papers need not be cited in later papers.
- For that matter you can consider that your main work product for the semester is one large report with several chapters

References

- References are outside sources cited or quoted in your paper
 - They may include both print and digital resources, preferably BOTH
- When you cite a reference, you typically make a general statement about some conclusion or statement of opinion in the reference – in your own words!
- It is far better to quote directly than to plagiarize by including or paraphrasing the author's words as if they were your own.

Annotated Bibliography

- In addition to including the formal citation in a standard format an annotated bibliography includes a sentence or two explaining exactly why the article or book is useful

Example

- Works Cited
- [1] E. Sapir, Dec. 1929, "The Status of Linguistics As A Science", *Language*, Vol. 5, No. 4 (Dec. 1929), pp. 207-214
Sapir's quote in the introduction of the paper is a well-earned way of explaining something of how a language influences the thoughts of those who use it: while he was referring to human languages, it holds true for programming languages as well.
 - [2] B. Eckel, *Thinking in Java*, Englewood Cliffs, New Jersey: Prentice-Hall, 2002.
Bruce Eckel's textbook is one of my go-to sources for a very high-level explanation of abstraction as it pertains to object-oriented programming in general.
 - [3] A. Kay, "The early history of Smalltalk", *ACM SIGPLAN Notices*, vol. 28, no. 3, Mar., pp. 2-54
As the developer of Smalltalk, Alan Kay laid out three primary points that have governed most of subsequent object-oriented programming languages' development.
 - [4] O. Dahl and K. Nygaard, "How Object-Oriented Programming Started", [Online document], available at: http://beim.ifi.sio.no/~kristen/FORSKNINGSDOK_MAPPE_F_OO_start.html
Ole-Johan Dahl and Kristen Nygaard were the creators of Simula. I can think of no better resource for discussion of the history of, and basic theory of, Simula.

The IEEE Citation Format

- The IEEE format is one of many
- Other popular formats are American Psychological Association (APA), Modern Language Association (MLA), etc.
 - Side note: whenever you use an acronym in a paper, spell it out on the first occurrence followed by the acronym in parentheses
 - Thereafter just use the acronym
- See Mayfield Ch 10

Elements of the IEEE Citation-Sequence System

- "Uses a single sequentially ordered note-number to cite all references to each source mentioned in the text." [2]
- Arranged by order of citation in the text (not alphabetically)

General Structure

- Citation numbers are enclosed in square brackets (ex: [1]) rather than superscripts¹
- IEEE encourages substituting reference numbers for the name of the author
 - ex: “[1], [9], and [11] have demonstrated...”
 - ex: “Use ‘in [1]’ instead of “in Williams et al.”
- Separation of elements: Except for a period ending the title element of a book, IEEE format uses commas rather than periods to separate elements.

General Structure

- Author names are in normal name order, usually with initials rather than first names. With multiple authors use a comma between all but the last pair of names
 - Ex. C. Meadow, L. Latour and P. Dickens
- Dates are given as dd mmm. Yyyy (when all parts are present)
 - Ex 9 Sep. 2009 Sep. 2009 2009
- Abbreviations, when used, are followed by a period.

General Structure

- Titles:
 - All significant words in book and journal titles are capitalized
 - Titles of journal articles and enclosed in quotes, or underlined, but only the first word is capitalized
- Page numbers are always preceded with p. or pp.

Examples (from [3])

- Books

C. W. Lander, *Power Electronics*, 3rd. ed., London: McGraw-Hill, 1993.

- Sections / chapters of books

G. K. Knopf and A. S. Bassi, “Biological-based optical sensors and transducers,” in *Opto-mechatronic Systems Handbook: Techniques and Applications*, Hyungsuck Cho, Ed. Boca Raton, FL: CRC Press, 2003, pp. 195-210.

Examples (from [3])

- Papers from Conferences

A. H. Cookson and B. O. Pedersen, “Thermal measurements in a 1200kV compressed gas insulated transmission line,” in *Seventh IEEE Power Engineering Society Transmission and Distribution Conference and Exposition*, 1979, pp. 163-167.

- Journal Articles

K. P. Dabke and K. M. Thomas, “Expert system guidance for library users,” *Library Hi Tech*, vol. 10, (1-2), pp. 53-60, 1992.

Examples (from [3])

- Online Journal Article

J. Farrell. (2007, May). In Wikipedia we trust? *Cosmos Online* [Online]. Available: <http://www.cosmosmagazine.com/node/1339>

- Online Conference Paper

X. Yang. (2003, Aug.). NIRA: a new Internet routing architecture. Presented at ACM SIGCOMM FDNA 2003 Workshop. [Online]. Available: <http://www.isi.edu/newarch/DOCUMENTS/yang.nira.pdf>

- Website (note unknown author)

(2007, Mar.). Dr Jean Armstrong: Brief Biography [Online]. Available: <http://www.ecse.monash.edu.au/staff/jeana/aboutarmstrong.html>

IEEE Style Guidelines

- Place note numbers directly after reference, not at the end of the paragraph (where superscript note numbers are often placed).
- In general, use the note number in place of author's name.

All languages have their limitations, however, as well as their strengths. Each has a dedicated area of focus, so to speak, where it is more effective at communicating an idea. In the world of spoken and written languages, English has a very detailed technical vocabulary and so even non-native speakers generally write and publish in the language. But, like all other languages, English has blind spots where it is analogous or less efficient at communicating an idea. The influence of language goes deeper, however, than one might think when looking at it on the surface, it strongly pushes the way that we think. [1] put it succinctly: "Human beings ... are very much at the mercy of the particular language which has become the medium of expression for their society. It is quite an illusion to imagine that one adjusts to reality essentially without the use of language and that language is merely an incidental means of solving specific problems of communication and reflection. The fact of the matter is that the 'real world' is to a large extent unconsciously built up on the language habits of the group."

IEEE Style Guidelines

- Use of note numbers sometimes seems a little strange. In this example the first occurrence of [3] could be replaced with a name:

Alan Kay, the designer and inventor of the Smalltalk programming language [3],...

[3] the designer and inventor of the Smalltalk programming language, laid out a few points that can be considered the primary features of an object oriented programming language. (It may be at first strange to reference [3]'s work, as Simula in particular predates Smalltalk by a number of years, but [3] merely codified what Simula began.) [3] actually wrote of six separate points, but only the first three he referred to could truly be considered universal to all OOP languages; the rest have been tinkered with heavily in many languages, including ActionScript.

IEEE Style Guidelines

- When more than one work is cited, separate note numbers with commas in one set of square brackets, e.g., [1,3,7]
- Never cite information that is not "recoverable" or viewable/obtainable by the reader
- Avoid content and additional bibliographic material such as you normally see in footnotes and endnotes – use parenthetical inserts instead
 - Ignore this rule for COS 301 as you will be preparing a non-standard annotated bibliography

Why All the Guidelines?

- It's the norm in academic publishing
- Citation formats are quite strict and are intended to provide a standardized set of data to readers

Appendices (sometimes Appendixes)

- "In one or more appendixes, include materials that are not essential parts of your main text but that will provide useful reference information to readers seeking more detail." [2]
- You can include:
 - Detailed Explanations
 - Additional Diagrams
 - Additional Tables
 - Long Lists
 - Program Listings relevant to the paper.

Appendices

- Avoid placing items with varying topics in the same appendix
- Label appendices with letters, e.g., Appendix A, Appendix B etc.
- Give each appendix a title and start on a new page

Appendix A
List of Types in Ada
...
Appendix B
Program Listing
- Refer to each appendix in the paper
- A paragraph of explanatory material may be added

Works Cited

- [1] R.P. Clark, *Writing Tools: 50 Essential Strategies for Every Writer*. New York, NY: Little, Brown, 2006
- [2] L. C. Perelman, J. Paradis, and E. Barrett, *The Mayfield Handbook of Technical & Scientific Writing*. Mountain View, CA: Mayfield Publishing Company, 1998
- [3] (2006, Apr.). Institute of Electrical and Electronics Engineers (IEEE) style examples. [Online]. Available: <http://www.lib.monash.edu.au/tutorials/citing/ieee.html>