

# Report Writing

(Get excited!)

**Abhiram Ranade**

Dept. of CSE, IIT Bombay

# Misconceptions about Writing

- Writing is important in Humanities but Not in Engineering or Science.
- Writing is an art. It is a divine gift. Unless you are born with it, or you cannot write well.
- Writing is easy. Good writers rattle of pages and pages overnight.
- There is no creativity in (technical) writing.
- Writing a document = giving all required information (i.e. core dump).

# Challenges of Good Writing

- Convey ideas, not just raw data. Ideas have structure: order, hierarchy.
- Convey what is important.
- Write compactly.
- Give good examples
- Reader should be able to understand something even if he/she “skims” through the document. No need to read from page 1 to page.
- Convenient organization/good chapter and section titles: Easy cross reference, Skimmers should be able to find relevant definitions without reading entire document.
- Should make the reader want to learn more.

# Outline

- Content of a report
- General Strategy
- Basic Principles of structuring text
- How to write a sentence
- Paragraph: the basic unit
- Lists, bullets..
- Introduction
- Structure of other chapters
- Pictures and examples
- Proofs, Algorithms, Citations

This talk is NOT about:

Latex, bibtex, xfig and other utilities. Learn these elsewhere.

# Content

Typical document has:

1. Problem Statement with motivation
2. Previous Work
3. Main presentation: in this the stated problem is solved or not.

# Problem Statement

If your audience does not understand the problem, it does not matter how good the rest of the talk is.

How to state the problem:

- Real life situation
- Input-Output: Most CS problems have input and output. State clearly. Make a distinction between database input and on-line input.
- Mathematical description of input and output, their representation, the constraints.
- Alternate formulations, relationships to other fields if relevant.

# Motivation

Why we need to do X

Why simple approaches do not work

Give people a reason to read your writing.

# Previous Work

Organize by ideas, not by author

# Main Content

**Some question is answered**

Question = problem, theorem, ...

Answer = Algorithm/Protocol/Formulation, Proof,  
Counter example

**Type of contribution**

Theoretical: Proofs

Experimental: Program runs. Careful selection on  
input

Claim either beauty or utility for your contribu-  
tion.

# Hard Results vs. Beauty

Look for and emphasize beauty

- Simplicity. Of two algorithms which are equally fast, the simpler one is considered more beautiful.
- Connect your work with different fields. Ad hoc algorithm vs. mathematically justified models..

# General Strategy: 1

## **Who are your readers?**

What do you need to explain? What is obvious?  
What is surprising to them?

Readers for MTech Seminar: Other MTech students and faculty, not necessarily specializing in the area of your seminar.

If you can satisfy a bigger population without being more verbose, *Great!*

## **Decide what MUST go into the report**

Make a list on a computer. “Core Dump”

Examples: Detailed examples are useful. Need creativity.

Pictures: worth thousands of words

## **Decide what to emphasize**

**Write in your own language.**

# General Strategy: 2

## Organize your material

Identify relationships between items: “A depends upon B”, “B and C are alternative ways of doing the same thing”, “D is more general than E”, “X and Y are similar except for ...”

Order ideas based on dependency.

Eliminate duplication: ideas that are common to different approaches should be discussed only once.

## Organizing Chapters:

Organize chapters by (i) ideas/intellectual questions (ii) Approach of XYZ, Approach of PQR... (i) preferred.

Ask yourself all the time: “Is this the best place for topic X, or is it better to put it after topic Y?”

**Rough draft:** Complete rough drafts of all chapters before generating final version of any chapter.

# Structure of Documents:

**Heirarchy of building blocks:** Chapters, sections, subsections, paragraphs, sentences.

**Begin each block with an overview:**

Report overview: first chapter

Chapter overview: first section.

Section overview: first paragraph

Paragraph overview: first sentence

**End each block with a summary**

Last chapter: Summary and Conclusions

Last section of each chapter: What did the chapter accomplish? What is important for the rest of the chapters.

...

**Exceptions to rule** Overview/summary not necessary if block is too small, or is exceedingly well written. “self-documenting code”

# Key idea in organizing a document

Match structure of ideas and of document

- Big Ideas: Chapters
- Small Ideas: Sections

One chapter (section) should contain only one big (small) idea.

**Mathematical Notation:** Important mathematical quantities should be given their special notation or mathematical symbols.

**Theorem/Lemmas/Definitions:** Reserve for important ideas. Less important ideas can go in paragraphs.

# Emphasizing an idea:

To emphasize an idea

- Give it its entire chapter (with suitable title)
- Give it its entire section (with suitable title)
- Give it its independent paragraph/several paragraphs.
- Define notation to express it.

*Converse also often holds..*

Direct method..

- Say so explicitly: “XYZ is extremely important in ...”

# How to write a sentence: 1

Same idea can be expressed by several sentences.

1. Cast iron is changed into steel when treated in a Bessemer converter.
2. When treated in a Bessemer converter, cast iron is changed to steel.
3. Cast iron, when treated in a Bessemer converter, is changed into steel.

Which sentence to use depends upon the context, and purpose.

Purpose = Essay on cast iron.

Context = Previous sentence talks about Bessemer converters

Purpose = Create suspense ... usually not needed in technical writing.

# How to write a sentence: 2

## Avoid unnecessary words

- “This is a subject which is liked by people”  
→ “This subject is liked by people”
- “Semaphores are used for synchronization purpose”  
→ “Semaphores are used for synchronization”
- “The logger program notes the fact that the event has happened”  
→ “The logger notes the event”

# Be Precise

“The problem as stated above is difficult.”

difficult for whom?

NP-complete? Believed to be NP-complete by experts (give reference)? Believed to be NP-complete by you?

# Paragraph: The Basic Building Block

**1 paragraph = 1 (atomic) idea/topic**

**Sentence 1 of paragraph:** Should suggest what the paragraph contains.

**Remaining sentences:** Elaboration of the subject idea

**Final Sentence:** Summary or statement of an important consequence of the topic of the paragraph. Or a transition.

The last possibility is particularly troublesome because there are so many ways in which you can crash your system if you have concurrent processes. What if you call `memchr` to search a block of memory that you have allocated and it garbles one of your memory manager's data structures? If a concurrent process—a code thread or an interrupt routine, say—then switches into context, it had better not invoke the memory manager because the system may crash. What if you call `memchr` to scan a global array and it steps on an adjacent variable used by another task? Or what if two instances of your program try to search shared data in parallel? Any number of scenarios can kill your program.

— Steve Maguire, *Writing Solid Code*

Just a few years ago people thought of computers as expensive and exotic devices. Their commercial and industrial uses affected ordinary people, but hardly anyone expected computers to become part of day-to-day life. This view has changed dramatically and rapidly as the public has come to accept the reality of the personal computer, small and inexpensive enough to take its place in every living room or even in every breast pocket. The appearance of the first rather primitive machines in this class was enough to catch the imagination of the journalists and produce a rash of speculative articles about life in the computer-rich world to come. The main subject of these articles was what people will be able to do with their computers. Most writers emphasized using computers for games, entertainment, income tax, electronic mail, shopping and banking. A few talked about the computer as a teaching machine.

— Seymour Papert, *Mindstorms*

# Lists

---

This program gives you a way of

- Presenting your material in the same manner as in a classroom.
  - Drawing squares.
  - Drawing circles.
  - Is better than similar programs.
- 

What is wrong with this?

All the items in a list should be of the same type, and of the same importance.

# The Introduction

Detailed overview of the report (5-6 pages)

Introduction  $\neq$  Abstract

---

## Chapter 1: Introduction

Light hearted opening

Statement of the importance of the problem

somewhat formal definition of the problem.

### 1.1 Overview

Overview of the rest of the chapter

### 1.2 ...

Summary of main problems and approaches

...

### 1.n Overview of Report

Chapter 2 discusses the xyz problem in more detail.

Chapter 3 discusses the pqr approach ...

---

Should entice reader to read more.

Give hints of key new ideas if possible.

# Structure of other chapters

---

## Chapter 2: Schemes for Representing PQR

light hearted comment about topic.

Formal statement of the problem addressed.

Relationship to the rest if not clear.

### 2.1 Overview

Summary of the rest of the chapter.

details about what is done in which section

### 2.2 ...

. . .

### 2.n Summary

---

# Writing Algorithms

**Describe overall idea and structure of the algorithm**

Try to convey intuition.

**Describe at a high level**

- “Insert  $x$  into the list  $y$ ” rather than code for insertion.
- “ $x = \text{sum of elements of } V$ ” rather than giving code.

**Explain data structures**

Mention invariants.

**Latex has packages for writing algorithms**

Manages indentation, highlights keywords.

# Terminology

All technical terms/notation must be defined very clearly, before they are used.

Use `\definition` environment in Latex.

Give chapter and section headings such that it is easy to find definitions.

# Credit Where Credit is Due

Very clearly mention the source of the ideas you discuss.

If an example is from a paper, mention that.

Same for figures!

Use `\cite` command liberally.

When citing URLs give dates.

## **Value addition by you:**

If some idea is your own, mention explicitly.

If you constructed an example, mention that.

If you found a shorter proof, mention that.

# Pictures and Examples

A picture is worth a thousand words.

Learn Xfig or whatever.

Use figure environment.

Examples: most necessary. This is where you can add value. Constructing good examples takes creativity.

Explain theorems with examples in addition to formal proof.

# Theorems and Proofs

Proofs must begin with an overview if they are long.

## **Bottom-up presentation**

Proof of theorem 3 uses proof of theorem 2.

Proof of theorem 2 uses theorem 3.

Conventional order.

## **Top-down presentation:**

Proof of Theorem 1 uses Theorem 2.

Proof of theorem 2 uses Theorem 3.

“Forward reference”. Provides motivation for reading theorem 2 etc. Acceptable.

# Concluding Remarks

- For perfection in writing: read and revise, again and again. Ask friends to read.
- Writing *is* an art: Rules are not absolute.
- Break rules if you must, but be prepared to say why.
- How to be a good writer: read a lot! Look for good writing style.
- Read Strunk and White's book – URL on my homepage.