Project Requirements (Cont)

- No copyright violations:
  - You need to re-draw all figures
  - You need to summarize all ideas in your *own* words
  - Cannot copy any part of text or figure unmodified
  - Short quotes ok
  - Any unmodified figures need permissions
- Any infringement will result in forfeiture of grades

Google Search (Cont)

- Google search, http://en.wikipedia.org/wiki/Google_Search
- How to search Google, http://www.wikihow.com/Search-Google
- Google Guide Quick reference: Google advance operators cheat sheet, http://www.googleguide.com/advanced_operators_reference.html
- Search Tips & Tricks –Inside Search – Google, http://www.google.com/insidesearch/tipstricks/all.html
- 12 Quick tips to search Google like an expert, http://blog.hubspot.com/blog/tabid/6307/bid/1264/12-Quick-Tips-To-SearchGoogle-Like-An-Expert.aspx
- Basic search help - web search help, http://www.google.com/support/websearch/bin/answer.py?hl=en&answer=34479&rd=1
- More search help - web search help, http://www.google.com/support/websearch/bin/answer.py?hl=en&answer=36861&topic=1221265
- Search results options, http://www.google.com/support/websearch/bin/answer.py?hl=en&answer=42143&topic=1221265
- Search preferences, http://www.google.com/support/websearch/bin/answer.py?hl=en&answer=35892&rd=1

Literature Search

- Conduct searches in two phases.
- In the first phase, use the title, key words of your project.
  - After reading these, conduct another more comprehensive search.
- Remove articles that are not useful
- No limit to the number of references
- Follow the references in references

Literature search

- Technical Papers,
- Industry Standards,
- White papers,
- Products,
- Web sites etc.

Outline Preparation Process

- Read abstract and stop if irrelevant
- Underline the key points in the paper
- Write the key summary on the first page of the paper
- Prepare a text document with the key ideas
- Keep adding to this text document from different papers with [refs, page, paragraph]
- Add figures and clean up the outline
- Like getting ready to make a presentation to the class

Use Mind Maps

Use Mind Maps

Use Mind Maps

One Clear Idea

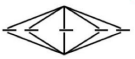
- Your paper should have just one "ping":
  - **one clear, sharp idea**
- Make **certain** that the reader is in no doubt what the idea is. Be 100% explicit:
  - "The main idea of this paper is...."
  - "In this section we present the main contributions of the paper."

Tell a story

Imagine you are explaining at a whiteboard

- Here is a problem
- It's an interesting problem
- It's an unsolved problem
- **Here is my idea**
- My idea works (details, data)
- Here's how my idea compares to other people's approaches

Diamond Writing Style



- Each paper should start with an introduction and end with a summary.
- Each section should start with a short introduction and end with a summary with a lead in to the next section. The same applies to subsections.
- All subsections should be of comparable length.
- Add an appendix with all abbreviations
- Add a list or discussion of related products

Writing Style

- Readers want to get to the information fast.
- Keep the nonessential stuff at the end.
- Check thoroughly for grammar and spelling.
- Avoid excessive use of abbreviations.
- Be consistent in case and usage: MOBILE, Mobile, mobile

Organization

- Every paper should have an introduction and a summary.
- Divide paper into sections.
- Every section should have a lead-in paragraph.
- Header level should correspond to the level in table of contents.

Organization (Cont)

- Title
- Author
- Abstract
- Keywords
- Table of Contents
- Introduction
- Other Sections
- Summary
- References
- List of Acronyms

Organization (Cont)

- Table of Contents
  - Sections and subsections
  - Numbering n.n
  - 3-7 subsections per section
  - 3-7 sections per paper
  - Include one first and 2nd level headers n. and n.n
  - Do not include 3rd and higher levels, e.g., n.n.n

Title/Keywords/Abstract

- Title
  - Based on Table of contents
  - Searchable
- Abstract
  - Based on Table of Contents
  - 3-7 sentences
  - Emphasize what part of course was used in the paper
- Key Search words
  - Based on Table of contents
  - Acronyms and full names
- Description:
  - One line based on ToC and Abstract

Organization (Cont)

- Introduction
  - Describe the problem
  - Why is important?
  - What is the main Idea?
  - Explicit – List of your Contributions
    - For each contribution refer to the corresponding section of the paper
    - The list of contributions drives the entire paper: the paper substantiates the claims you have made

Presenting the idea

- Explain it as if you were speaking to someone using a whiteboard
- **Conveying the intuition is primary,** not secondary
- Once your reader has the intuition, she can follow the details (but not vice versa)
- Even if she skips the details, she still takes away something valuable

Conveying the intuition

- Introduce the problem, and your idea, using **EXAMPLES** and only then present the general case
- Remember: explain it as if you were speaking to someone using a whiteboard

Example

Which of the two is best in practice? The trouble is that the evaluation model has a pervasive effect on the implementation, so it is too much work to implement both and pick the best. Historically, compilers for strict languages (using call-by-value) have tended to use eval/apply, while those for lazy languages (using call-by-need) have often used push/enter, but this is 90% historical accident — either approach will work in both settings. In practice, implementors choose one of the two approaches based on a qualitative assessment of the trade-offs. In this paper we put the choice on a firmer basis:

- We explain precisely what the two models are, in a common notational framework (Section 4). Surprisingly, this has not been done before.
- The choice of evaluation model affects many other design choices in subtle but pervasive ways. We identify and discuss these effects in Sections 5 and 6, and contrast them in Section 7. There are lots of nitty-gritty details here, for which we make no apology — they were far from obvious to us, and articulating these details is one of our main contributions.

In terms of its impact on compiler and run-time system complexity, eval/apply seems decisively superior, principally because push/enter requires a stack like no other: stack-walking

Other Sections

- Each section less than 3 pages.
- Each section needs at least one introductory paragraph. Do not start with a subsection.
- Each subsection at least 1 paragraph.
- All sections/subsections should be numbered n. n.n
- If you borrowed several sentences from some source, italicize the text and indicate the source.
- Never write a sentence with the original source in front of you. This will block you from writing it in your own words. Write ideas from source, merge ideas from different sources, and then write the ideas in your own words.

Other Sections (Cont)

- Always include citations for sources of ideas even if the expression is yours.
- Redraw the line drawings. Avoid copy and paste as much as possible (e.g., for photographs).
- If there is no author, e.g., a web page, use the site name followed by a number, e.g., [wikipedia02] in the body of the text, with full title and url in the reference list.

Summary

- One or more sentences about the each issue.
- Based on Table of Contents
- Key lessons

References

- Style of References
  - Author(s), "Title," Source, date, pages, url
  - The URL should show up as well as have a link.
- Find URL for published papers
- Remove references that are useless.
- Exception: Standards, company documents, RFCs.

List of Acronyms

- Search the text
- Define on first use
- Avoid acronym use if used less than 5 times.
- Exception: Commonly used acronyms, e.g., CPU, I/O, IP, …

Figures/Tables

- All figures should be numbered 1, 2, …
- All tables should be numbered 1, 2, …
- All figures should have a title below the figure
- All tables should have a title above the table
- All figures/tables should be referenced in the text and explained.
- Should be placed close to their references.
- To prepare figures in Windows use blank slides in PowerPoint using the font sizes to be used in the paper. Group the figure as one object. Copy and use paste special as "Enhanced Windows Metafile (EMF)" for best quality.
- Do not rescale figures in Word. Rescaling the fonts reduces their quality.

Editorial

- Check all acronyms. All acronyms should be defined on first use.
- Check capitalization. No unnecessary capitalization. Headers are usually capitalized.
- Spell Check entire document.
- K=1024, k=1000. Disk storage is measured in KB, network link speeds are measured in kb. kbps not Kbps.
- Leave a space between numbers and units, e.g., 15 km not 15km.

Visual structure

- Give strong visual structure to your paper using
  - sections and sub-sections
  - bullets
  - italics
  - laid-out code
- Find out how to draw pictures, and use them

Example

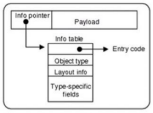


Figure 3. A heap object

The three cases above do not exhaust the possible forms of f. It might also be a *THUNK*, but we have already dealt with that case (rule THUNK). It might be a *CON*, in which case there cannot be any pending arguments on the stack, and rules UPDATE or RET apply.

4.3 The eval/apply model

The last block of Figure 2 shows how the eval/apply model deals with function application. The first three rules all deal with the case of a *FUN* applied to some arguments:

- If there are exactly the right number of arguments, we behave exactly like rule KNOWNCALL, by tail-calling the function. Rule EXACT is still necessary — and indeed has a direct counterpart in the implementation — because the function might not be statically known.
- If there are too many arguments, rule CALLK pushes a *call* remainder of the object is called the *payload*, and may consist of a mixture of pointers and non-pointers. For example, the object $CON(C\ a_1 \ldots a_n)$ would be represented by an object whose info pointer represented the constructor C and whose payload is the arguments $a_1 \ldots a_n$.

The info table contains:

- Executable code for the object. For example, a *FUN* object has code for the function body.
- An object-type field, which distinguishes the various kinds of objects (*FUN*, *PAP*, *CON* etc) from each other.
- Layout information for garbage collection purposes, which describes the size and layout of the payload. By "layout" we mean which fields contain pointers and which contain non-pointers, information that is essential for accurate garbage collection.
- Type-specific information, which varies depending on the object type. For example, a *FUN* object contains its arity; a *CON* object contains its constructor tag, a small integer that distinguishes the different constructors of a data type; and so on.

In the case of a PAP, the size of the object is not fixed by its info table; instead, its size is stored in the object itself. The layout of its fields (e.g. which are pointers) is described by the (initial segment of) an argument-description field in the info table of the FUN object which is always the first field of a PAP. The other kinds of heap object all have a size that is statically fixed by their info table.

A very common operation is to jump to the entry code for the object, so GHC uses a slightly-optimised version of the representation in Figure 3. GHC places the info table at the addresses *immediately*

Editorial (Cont)

- Look for special characters
- Check for continuity
- Break long paragraphs.
- Single space between paragraphs.
- The paper should be 8-10 pages long
- If you copy any figures, give reference and credit
- Use the template supplied

Use the active voice

No!	Yes!
It can be seen that...	We can see that...
34 tests were run	We ran 34 tests
These properties were thought desirable	We wanted to retain these properties
It might be thought that this would be a type error	You might think this would be a type error

Use simple, direct language

No!	Yes!
The object under study was displaced horizontally	The ball moved sideways
On an annual basis	Yearly
Endeavour to ascertain	Find out
It could be considered that the speed of storage reclamation left something to be desired	The garbage collector was really slow

Common Mistakes

- No Figures
- Figure/equations fonts too large
- Figures with no title or number or reference
- Figures/tables overflowing the margins
- References with no annotation
- References not cited
- Key pieces of information w/o references
- Tables w/o references
- Papers too short

Common Mistakes (Cont)

- No comparison of different alternatives
- No Acronyms
- No summary
- Incorrect reference style
- No keywords

Checklist

1 Are Keywords appropriate?
2. Is the title satisfactory?
3. Does the abstract clearly summarize the topic discussed?
4. Table of Contents logically organized?
5. Does introduction entice you to read the rest of the paper?
6. Major ideas and topics received enough attention?
7. Are individual sections and subsections of uniform length?
8. Are references correctly formatted and spread throughout?
9. Include author, title, dates, pages, and URL?
10. Did the author follow the diamond explanation principle?
11. Acronyms used properly and listed?
12.Figures and Tables (Clearly labeled and professional looking, referenced in the text and explained)
13. Are paragraphs of right length (not too long or too short)?
14. Do the subheadings clarify the sections of the text?
15. Was the material ordered in a way that was logical, clear, and easy to follow?
16. Is there any portion of the text that could be omitted?
17. Does the summary point out the key results?
18. Copyright violations in text, figures, or tables?
19. Text checked for Grammar, Spelling, Punctuation errors