Brief Midterm Feedback & Discussion

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

Letter	Percent Range	Real Grade Range	Highest Percent	Highest Real Grade	Lowest Percent	Lowest Real Grade	Percent Average	Real Grade Average	Percent of Students	# of Students
A	92.5% - 100.0%	92.5 - 100.0	100%	100	93%	93	96.3%	96.3	32.4%	24
A-	89.5% - 92.4%	89.5 - 92.4	92%	92	90%	90	90.7%	90.7	12.2%	9
B+	86.5% - 89.4%	86.5 - 89.4	89%	89	87%	87	87.9%	87.9	9.5%	7
B	82.5% - 86.4%	82.5 - 86.4	86%	86	83%	83	84%	84	12.2%	9
B-	79.5% - 82.4%	79.5 - 82.4	82%	82	80%	80	80.6%	80.6	10.8%	8
C+	76.5% - 79.4%	76.5 - 79.4	79%	79	78%	78	78.3%	78.3	4.1%	3
C	72.5% - 76.4%	72.5 - 76.4	76%	76	73%	73	74.6%	74.6	12.2%	9
C-	69.5% - 72.4%	69.5 - 72.4	71%	71	71%	71	71%	71	1.4%	1
D+	66.5% - 69.4%	66.5 - 69.4	69%	69	68%	68	68.5%	68.5	2.7%	2
D	59.5% - 66.4%	59.5 - 66.4	66%	66	60%	60	63%	63	2.7%	2
F	0.0% - 59.4%	0.0 - 59.4	0%	0	0%	100	0%	0	0%	0
Total										**74**

Quick Review of Specific Questions on the Midterm

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

Iteration #2 (FlashPhoto) Q & A:

Any questions to follow up the presentation by TAs on Tuesday?

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

Writing Techniques for Software Development: Writing Effective Project Reports

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

Common Scenarios

- Update other programmers and/or managers (better for us to practice because it requires some more formalism/responsibility) on technical progress.

- Update a client on technical project status.

From Your Assignment Posted on Moodle

The **purpose** of both emails is to provide a status update on your project. This includes what has been accomplished to date and what is planned to be accomplished next.

The **audience** for email #1 is a technical manager for your project.

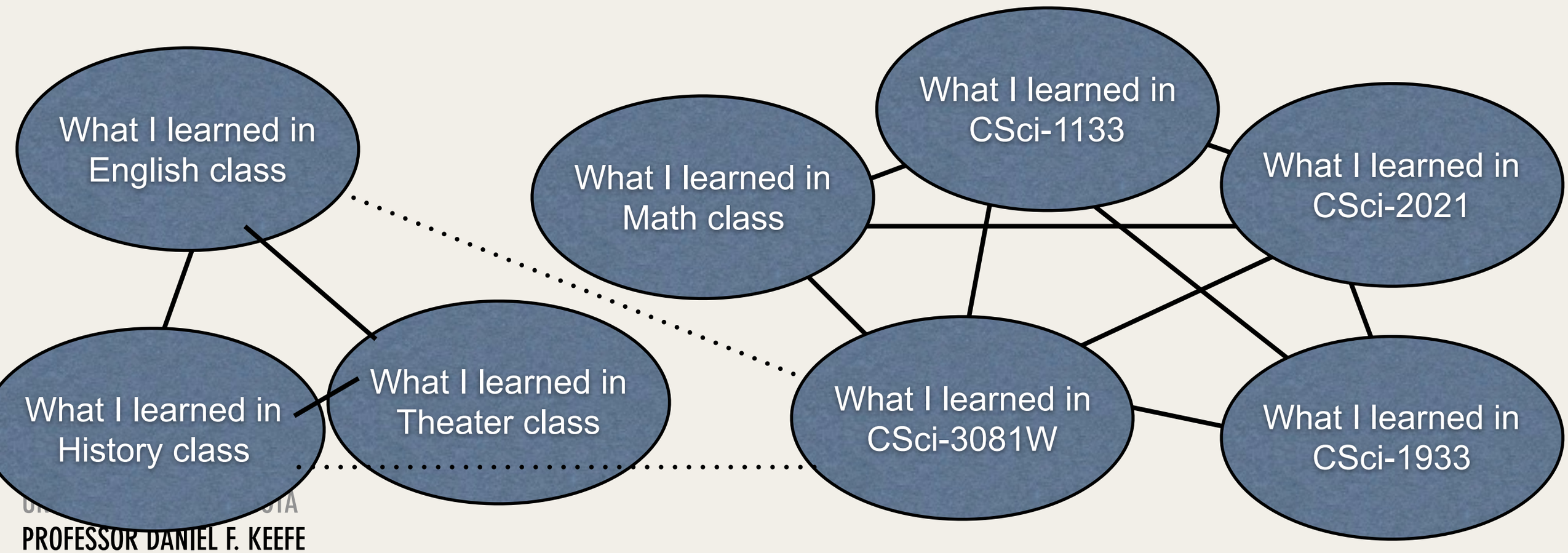
The **audience** for email #2 is the client sponsoring our project.

From My Experience

- All software development projects require some sort of report, it could be monthly, quarterly, annually, and/or at the end of the project.
- As part of this (and often in addition) frequent email communication with a client is typically required. If the client is a company, they might assign a contact person for this email communication.
- Often that contact person is an expert in his/her field and may be technically oriented, but it's rare that this person is actually a programmer.
- This means that the contact person (and by extension your client) can sometimes have no idea how long it will take to implement various features and how challenging this will be for you to do.
- The reports/emails become a critical way to communicate about this and manage it. For example, some clients might push for you to do more, more, more… Rather than saying "no" (or worse saying "yes") to these requests, you need to explain/teach the client about what you have done, why it was challenging, what surprises have come up and how they will be handled, etc.

What makes this so difficult for students to learn?

- A lack of familiarity with examples of good technical writing.
- A lack of practice with technical writing.
- My hypothesis: A disconnect between knowledge of writing and knowledge of programming.



These difficulties lead to two common pitfalls in project reports/updates:

1. Writing an overly generic project report that is full of platitudes and generalizations (e.g., everything is on schedule, all tasks have been successfully completed) so that the report basically contains no real information and is hard to believe.

2. Making up a story that is not actually accurate.

So, what's the right way to do this type of writing?

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

First, break the task into component steps:

1. Identify what you did.

2. Out of these tasks that you did, identify what tasks are “important”, where important can mean different things (e.g., a challenge overcome, a challenge not overcome, a surprise roadblock). Note, important does not necessarily mean successful.

3. Identify how the audience that you are writing to will react to these important things. Will the audience understand and agree that they are important?

4. Select an appropriate set of these things to include in your report. You certainly can include things that the audience will not understand or agree are important, but you’ll need to explain and justify these as part of the report.

5. Write about these things using language to describe them that the audience will understand.

Second, put extra effort into identifying good details to include in your report.

- Focus on what you think is important, don't worry to start with whether it is a "good thing" or a "bad thing".
- Your instinct might be to avoid talking about something that you struggled with because it might make you look bad.
  - Turn this "negative" into a positive. No project is perfect, so it's expected to have hurdles and bumps in the road. If you don't describe any challenges, then people might think they are paying you too much for such an "easy project" :)
  - Convincingly describe the challenge and how overcoming it or having a plan to overcome it now puts the project in a good situation.
- Get specific whenever you can: mention features by name, list tasks completed rather than saying "all tasks are complete", give numbers when you can (major classes, lines of code, developer hours devoted to the project, …).

A Real-ish Project Update Email Example

Monthly Summary Report: March 1, 2010

Name of Project…
Report Period: February 1, 2010 to February 28, 2010
Names of Project Team Members…
University of Minnesota

1. *Summary of Progress Since Last Report*

- Major accomplishment: Detailed planning for software implementation for all Aims. [It's early in the project, so this focus on software planning makes sense.]
- Additional planning for software implementation for new features brainstormed in the recent face-to-face meeting. [Would be better to add detail here on how this planning was done, e.g., formal UML diagrams?]
- In all, more than 40 software features analyzed and discussed enough to form an implementation plan and estimate programming effort required in hours of work time. [Great use of a quantitative measure of success — 40]
- All this is reported in an 11 page document, the first deliverable of the project. [Again, found a way to insert a number into the report to add detail!]
- The document outlines a proposed software release schedule with specific features tied to a series of 3 planned software releases during the next year. [Again, note "3 planned software releases during the next year" — technically this information is not "required", the sentence would still work without it, but this adds technical detail that even a non-technical client can understand, and this gives the report credibility.]
- Software development is on schedule with the proposed release plan (see progress specifics below). However, we intend for the release schedule to be able to be determined through collaborative discussion. [The first sentence is generic, but this section of the email report is just a high-level summary of progress, so it's ok here. Notice, it points to more detailed info to follow. The second sentence just acknowledges that the client will have some say helping to define future software features — this may or may not be the case depending upon the particular project.]

2. *Work Progress for Each Aim*

Aim 1: A descriptive name for this aim.
- Developed detailed software specification for all feature requests.
- The [X] and [Y] features were particularly challenging in this regard. Current computing hardware requires [insert some more detail] and this makes doing [X] together with [Y] a significant technical challenge. The design developed overcomes this challenge through a creative use of [Z].
- To date, all programming effort has focused solely on this Aim. [This is fine, it's early in the project and this gives the audience an understanding that these things take time.]
- Specific features implemented to date include [XXX].
- These features have been tested using a [YYY] methodology that is designed to extend naturally as the implementation proceeds in the remainder of the project.

Aim 2: Name of the aim.
- Developed detailed software specification for all feature requests.
- Developed user experience guidelines and user interaction storyboards for the most critical features within this Aim, including [ZZZ].

[More detail…]

Aim 3: Name of the aim.
- Developed detailed software specification for all feature requests.

[Less detail here is ok at an early stage in the project…]

Monthly Summary Report: March 1, 2010

Name of Project…
Report Period: February 1, 2010 to February 28, 2010
Names of Project Team Members…
University of Minnesota

1. Summary of Progress Since Last Report

- Major accomplishment: Detailed planning for software implementation for all Aims. [It's early in the project, so this focus on software planning makes sense.]
- Additional planning for software implementation for new features brainstormed in the recent face-to-face meeting. [Would be better to add detail here on how this planning was done, e.g., formal UML diagrams?]
- In all, more than 40 software features analyzed and discussed enough to form an implementation plan and estimate programming effort required in hours of work time. [Great use of a quantitative measure of success — 40]
- All this is reported in an 11 page document, the first deliverable of the project. [Again, found a way to insert a number into the report to add detail!]
- The document outlines a proposed software release schedule with specific features tied to a series of 3 planned software releases during the next year. [Again, note "3 planned software releases during the next year" — technically this information is not "required", the sentence would still work without it, but this adds technical detail that even a non-technical client can understand, and this gives the report credibility.]
- Software development is on schedule with the proposed release plan (see progress specifics below). However, we intend for the release schedule to be able to be determined through collaborative discussion. [The first sentence is generic, but this section of the email report is just a high-level summary of progress, so it's ok here. Notice, it points to more detailed info to follow. The second sentence just acknowledges that the client will have some say helping to define future software features — this may or may not be the case depending upon the particular project.]

2. Work Progress for Each Aim

Aim 1: A descriptive name for this aim.

- Developed detailed software specification for all feature requests.
- The [X] and [Y] features were particularly challenging in this regard. Current computing hardware

planned software releases during the next year. [Again, note "3 planned software releases during the next year" — technically this information is not "required", the sentence would still work without it, but this adds technical detail that even a non-technical client can understand, and this gives the report credibility.]

- Software development is on schedule with the proposed release plan (see progress specifics below). However, we intend for the release schedule to be able to be determined through collaborative discussion. [The first sentence is generic, but this section of the email report is just a high-level summary of progress, so it's ok here. Notice, it points to more detailed info to follow. The second sentence just acknowledges that the client will have some say helping to define future software features — this may or may not be the case depending upon the particular project.]

2. Work Progress for Each Aim

Aim 1: A descriptive name for this aim.

- Developed detailed software specification for all feature requests.
- The [X] and [Y] features were particularly challenging in this regard. Current computing hardware requires [insert some more detail] and this makes doing [X] together with [Y] a significant technical challenge. The design developed overcomes this challenge through a creative use of [Z].
- To date, all programming effort has focused solely on this Aim. [This is fine, it's early in the project and this gives the audience an understanding that these things take time.]
- Specific features implemented to date include [XXX].
- These features have been tested using a [YYY] methodology that is designed to extend naturally as the implementation proceeds in the remainder of the project.

Aim 2: Name of the aim.

- Developed detailed software specification for all feature requests.
- Developed user experience guidelines and user interaction storyboards for the most critical features within this Aim, including [ZZZ].

[More detail…]

Aim 3: Name of the aim.

- Developed detailed software specification for all feature requests.

[Less detail here is ok at an early stage in the project…]

Where is our class going with these?

- We’re going to put some effort into these. These project reports are a focus for our class. You all need to learn how to write these effectively.

- This week, you’ll create a first draft.
- Next week, you’ll write a “peer response” to another student’s reports.
- The following week, you’ll get to read a helpful peer response from another student and you’ll then revise your emails based upon it.