CSci-3081W: Program Design and Development

Professor Daniel Keefe
Spring 2015

Who am I really?

Safi, 3

Aidan, 8

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

Two Super TAs

Dan Orban

Email: dtorban@umn.edu

Office Hours: During the Friday lab sessions or by appointment.

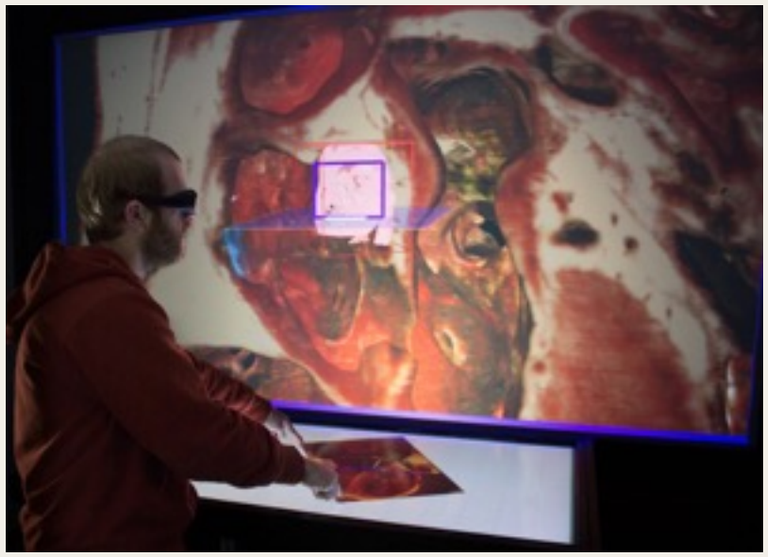
Kevin Thomsen

Email: thoms230@umn.edu

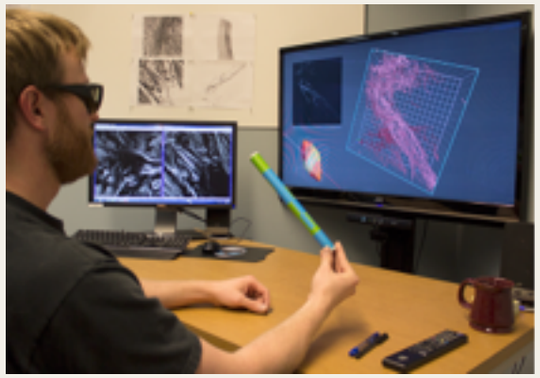
Office Hours: During the Friday lab sessions or by appointment.

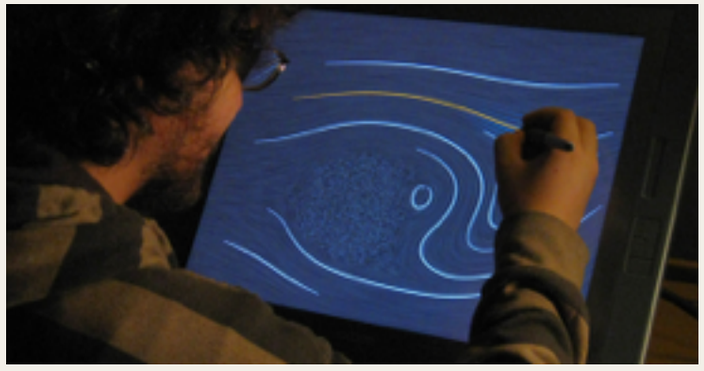
Why am I excited about this course?

CavePainting & SciVis Applications
Brown University, 2001-2007



SliceWIM, etc.
Univ. of Minnesota
2008-Present





Jumping Right In: Let's Look at Some Software

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

Adobe Photoshop Elements

https://www.youtube.com/watch?v=cicRy_pbb1A

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

Imagine you are developing this software… What would make this program challenging to design and develop?

- Example: The need for 3 different interfaces (Quick, Guided, and Expert) for *every* feature!

Imagine Adobe hired your consulting business for some outside programming help, and they want you to add a new filter to Photoshop Elements…

- In a well designed program, would this be easy to do?
- Can you think of a good design strategy that would make this easy?
- What about a bad design strategy that would make it really hard to make a change like this?

Paper

https://www.youtube.com/watch?v=DiDLgLKB3yw

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

The designers of Paper made some design decisions that are really different from Photoshop Elements.

What makes Paper successful?

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

CSci-3081W - Program Design and Development: Introduction and Mechanics of the Course

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

Your Learning Goals

- Learn skills, tools, and theory related to becoming a good software developer.
- Prepare to succeed in 4xxx and 5xxx programming-intensive classes.
- Prepare to succeed in “real world” programming projects in industry to graduate school after you graduate.

Some Specific Topics

- The software engineering process
- C++ programming
- UML diagramming
- Software development tools (Makefiles, debuggers, version control)
- Good coding practices
- Professional skills, especially related to writing and team-based programming

Prereqs

- Officially: CSci (1902, 1913, or 1933) and CSci 2021

- Essentially: Prior programming experience with C, C++, Java, or a similar language. *(Without this you would likely find the jump to serious C++ programming a bit too challenging.)*

- Also: Experience with basic data structures (lists), algorithms (search), recursion, and data abstraction.

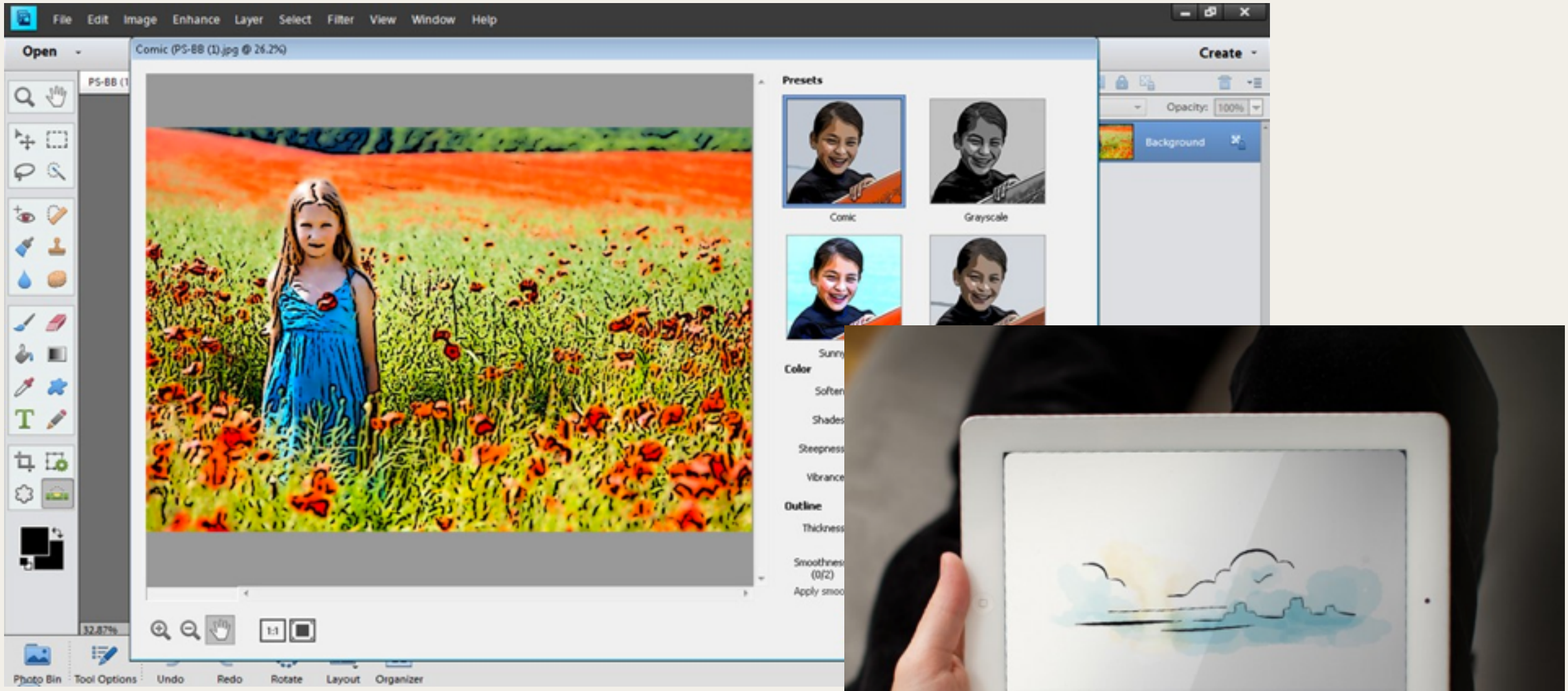
- Speak with me or send email if you have questions on this.

Course Structure

A "Big" Program Design and Development Project

- Working as a team with other programmers
- Takes months to complete
- Requires ~100 files of source code and tests
- Links with multiple external libraries
- Requires use of good programming practices

Our Project Topic



UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

All Components of the Course are Organized around the Project

- Labs
- Weekly Writings
- Semester Schedule
- Group Work
- …

Labs: Key Points

- Register for one (if not already)
- Two purposes:
  - TA-led Instructional Labs: TAs will distribute technical information on the project, lead “help sessions”, etc. **(Attendance Required)**
  - Project Groupwork Labs: You will meet regularly with your group to organize project work.
- No Lab meeting this week, we will start next Friday.

Weekly Writings: Key Points

- Program Design and Development requires a lot of writing:
  - Writing solid, self-documenting code
  - Drawing design diagrams
  - Documenting intended use of code
  - Reporting on development progress
- You are going to learn and practice all of these.
- For the first half of the semester, you will practice these forms of writing with weekly individual short writing assignments.

Semester Schedule and Project Iterations

- **PART 1:** ***"The Training Stage"***
  - Individual: Weekly Writing assignments throughout
  - Group: Iteration #1 of the Project (Digital Painting)
  - Group: Iteration #2 of the Project (Photo Editing)

- **PART 2:** ***"The Guru Stage"***
  - Group: Iteration #3 of the Project (Project Release)
  - More challenging programming, writing as a group to facilitate a "project release", apply what you learned in part 1.

Course Webpage

- Let's go there now…
- https://ay15.moodle.umn.edu/enrol/users.php?id=12080

- This is absolutely essential to the success of the course.
- It is going to fill in and change as we get going with the course. Check it daily.
- Look for homework assignments and lab schedule.

Team-Based Programming: Motivation and Approach

- All serious programming “in the wild” is team based.
- Learning to write code that works as part of a big project with multiple programmers takes practice.
- You’ll fill out a “Getting to Know You” questionnaire. We’ll use this to help us assign you to a team of 3 students, all in the same lab section. This will be your project team for the whole semester.
- In class and in labs, you will learn a number of strategies for facilitating team-based programming and teamwork in general, including creating team policies and assigning roles and responsibilities.

Team-Based Programming: Some Policies

- We recommend that you each take on different roles, but every member must contribute (approximately equally) to each group assignment.
- Every member of a group will receive the same grade for assignments completed as a group.
- We reserve the right to disband a group if it is dysfunctional. (This is extremely rare.)
- If you group isn't working, the first thing to do is discuss the situation with your group, referring back to the group policies and expectations that you established. If this doesn't work, then discuss the situation with me, and do this early enough so that there is still time for me to do something about it!

Another Very Important Class Policy: NO LATE WORK

- We will share our solutions to assignments (including project code) immediately after each assignment is due.

Almost Done…

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

Assessment: Final Grades

- 6% Labs and Small Exercises (In and Out of Class)
- 20% Individual “Weekly Writing” Assignments
- 13% Group Project Iteration #1
- 13% Group Project Iteration #2
- 18% Group Project Iteration #3
- 10% Midterm
- 20% Final Exam (cumulative)

Academic Integrity

- #1 Rule: *"Do your own Program Design and Development"*
- Includes your own: *Thinking, Design, and Coding.*
- Use of the Web:
  - OK: As a supplement to the textbook, as a resource to understand a strange compiler error, …
  - NOT OK: As a source for the *solution* for an assignment, e.g., Google: "how to implement a paint program in C++".
  - It's very important to me that this is clear. Read the syllabus for more specific examples. Just ask me first if you need clarification at any time during the semester.

Any Questions on Course Structure & Policies?

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE

Ok, We’re Ready to Start!

Let’s check the webpage to see what to do before our next meeting…

UNIVERSITY OF MINNESOTA
PROFESSOR DANIEL F. KEEFE