# Nifty Assignments and Tools Panel

Stuart Hansen (moderator)
Department of Computer Science
UW - Parkside
Kenosha, WI 53144
hansen@uwp.edu

Teresa Nickeson
Computer Science and Mathematics Dept.
University of Dubuque
Dubuque, Iowa 52001
TNickeson@dbq.edu

Kristine J. Peters
Mathematics and Computer Science
Ripon College
Ripon, WI 54971
petersk@ripon.edu

Tim Gegg-Harrison and Nicole Anderson
Department of Computer Science
Winona State University
Winona, MN 55987
{tgeggharrison,nanderson}@winona.edu

David R. Musicant
Department of Computer Science
Carleton College
Northfield, MN 55057
dmusican@careleton.edu

Dean Stevens and Randy Campbell
Mathematical Sciences Department
Morningside College
Sioux City, Iowa 51106
{stevens,campebell}@morningside.edu

Samantha S. Foley
Department of Computer Science
UW – La Crosse
La Crosse, Wi 54601
ssfoley@cs.uwlax.edu

## Abstract

This panel will present a series of nifty assignments and tools. Each presenter has a particularly successful assignment and tool. The assignments are successful because they do a great job of illustrating some concept, or because the students can thoroughly relate to them, or some combination of both. The tools may be new and interesting IDEs, improved content management systems, or specialized tools for particular subjects. All contribute to improving our courses, engaging students and promoting learning. In other words they are nifty.

# 1 System Analysis & Design In-Class Group Project

Teresa Nickeson

## 1.1 Abstract

The assignment I have chosen for the Nifty Assignment Panel Discussion is an in-class assignment I use with the Systems Analysis and Design chapter of my Intro to CIS survey class. Instead of a tradition lecture, students actively work through the phases of the Systems Development Life Cycle (SDLC) in groups. They are given a scenario and asked to detail the tasks they would perform in their phase.

## 1.2 Introduction

Intro to CIS is the first class for any of the University of Dubuque's computer majors. It covers twelve computer-related topics, one of which is Systems Analysis and Design. When students read the book, it all seems quite logical and sensible. However, when they are asked to role play this work, new dimensions of learning occur.

## 1.3 The Assignment

The students are divided into five teams, one for each of the phases of the Systems Development Life Cycle (SDLC). The class is given a business project scenario. Each group is asked what activities would be performed in their phase of the process specific to the project. Students are to assume that the project has been approved at the end of the phases before theirs. They can make any assumptions they want about the conclusions the upstream phases reached.

As an example of the detail desired, in the investigation phase, students would not just note that they would prepare a preliminary cost / benefit analysis; they would contemplate what the costs and benefits might be. Instead of just noting that they would complete feasibility studies, they would figure out which would be germane for the project given and what should be considered for each.

When finished, the groups informally present their work to the class.

## 1.4 The Experience

Students usually complain that they do not have enough information in my short project overview to get started. That leads to a discussion about how in the real world you have to ask these questions.

Most groups cannot resist the temptation to go beyond their phase to begin designing the solution. This leads to a good discussion about the cost of "runaway" analysis and the approvals that are usually needed to move from one phase to another. Students then regroup to figure out what turnover documents might be appropriate for each phase and who should approve them.

Students discover that it is not always clear which specific tasks go in each phase. This illustrates the point every author and professional will make about the importance of

communication and working together. It also illustrates that the phases do not necessarily have rigid lines and that the phases can vary some by project.
The presentations spark good questions, especially from the downstream groups which may have made other assumptions. For example, the Implementation Phase team may wonder why the Design Phase team decided to purchase instead of build the solution.

## 1.5 Conclusion

I could lecture about all of the SDLC phases and activities, but it would not be as effective as what the students retain “experiencing” the phases and coming up pertinent questions. The points I do make after each group discusses their finding make more sense, too, in the context of the role playing students just completed.

## 1.6 Biographical Information

Teresa Nickeson is an associate professor of Computer Information Systems and the head of the Computer Science and Mathematics department at the University of Dubuque. She has 20+ years of industry experience with McGraw-Hill leaving as the VP of Sales and Marketing Applications. She has a B.S. of Computer Science and Math from UW-Platteville and an MBA from the University of Dubuque.

# 2 Using Gamification to Engage Students While Learning Mathematical Induction

Tim Gegg-Harrison and Nicole Anderson

## 2.1 Abstract

While students are often intimidated by discrete mathematics concepts such as proof by induction, they are often drawn to games and specifically games on mobile devices. The turn-based nature of many mobile apps also provides a social connection with their classmates, allowing students an interactive learning experience. We have constructed a mobile app to be used concurrently with the presentation of the concept of mathematical induction in a discrete mathematics course. The app allows students to play a turn-based strategy game that captures the subtraction game variant of *Nim*, where players take turns removing objects from a pile and the player that removes the last object loses. Strategy is clearly important and devising an optimal strategy is possible and can be proved using mathematical induction. Our objective is to use the game side-by-side with a proof by induction to introduce the concept of mathematical induction in a fun and engaging manner.

The material for this paper was created for a discrete mathematics course (CS 275 *Mathematical Foundations of Algorithms* at Winona State University) and is available at http://tiniapps.com/education/MIExercise/index.xhtml.

A more complete description of this project may be found elsewhere in the proceedings.

## 2.2 Biographical Information

Tim Gegg-Harrison and Nicole Anderson are professors in the Computer Science Department at Winona State University. Their teaching and research interests include CS education, mobile app development, and discrete mathematics.

# 3 The Reserved Word Program
Kristine Peters

## 3.1 Abstract

In this series of assignments, students are to find the identifiers in a C++ source code file, either counting how often the "reserved words" or keywords appear or, in an advanced course, building a tree of the non-keywords.

## 3.2 A Series of Assignments

In Computer Science I, the students are to find and count the frequency of the C++ keywords in a source code file. Information about the keywords is stored in an array of structures; each entry contains a keyword and an integer representing its frequency in the file. As each string is read from the file, the alphabetic characters are translated to lowercase and a binary search is used to determine if the string is one of the C++ keywords.

In Computer Science II, the assignment is extended to add a linked list of line numbers in which the keyword is found to each keyword structure.

The more interesting part comes in its reincarnation in Data Structures and Algorithms. Instead of counting keywords, the students are to create a listing of all the identifiers found in a C/C++ source program. The identifiers are stored in a binary search tree, along with a frequency counter and a linked list of all the line numbers in which the identifier is found. Again, a binary search is used to eliminate the keywords found.

An extension to the BST version is, once the BST has been created, to add a menu allowing the user to find specific identifiers and output their line numbers, delete an entire entry (including the linked list) for a specific identifier, add new identifiers and corresponding line numbers, and print the entire tree.

Over three courses, these assignments develop the students' use of an array of simple structures, an array of structures with a linked list as one of the fields, and a binary search tree. String processing and binary search techniques are used in each assignment. It illustrates an essential feature of compilers, finding identifiers and storing information about them. It allows the students to modify code that they have previously created.

## 3.3 Biographical Information

Kristine Peters is an Associate Professor of Mathematics and Computer Science at Ripon College where she has received the Severy Award for Teaching Excellence two times. She studied at the Ohio State University and Purdue University. Her teaching interests include: data structures and algorithms, programming languages and operating systems.

# 4 Wikipedia Article Gender Analysis
# David R. Musicant

## 4.1 Introduction

All of our students are familiar with Wikipedia, which is an amazing community-driven project where anyone can make edits to almost any article. One key issue with Wikipedia that was identified by the New York Times [2] is that only 13% of contributors to it are women. This would obviously seem to be a problem, though skeptics might argue that the gender of an author doesn't matter if the content of the articles themselves well represent the interests of all. Can one look at Wikipedia and determine if its *content* is biased?

## 4.2 The Assignment

This assignment invites students within a CS 1 class to reproduce research [3] that looked at English Wikipedia articles of interest to women and articles of interest to men, and compared the lengths of these articles. Interestingly, it has been shown that Wikipedia article length correlates well with article quality [1,4].

This nifty assignment consists of both code writing as well as document writing. Students write code to determine some experimental results, but they then write up their results in a short report. This assignment was created in order to help integrate writing into a CS 1 class; additionally, it helps CS 1 students get a taste for doing research in social computing. Furthermore, it helps illustrate how computer science can be used to gain access to an elusive question, namely as to whether or not a gender bias exists within Wikipedia content.

Determining the (approximate) length of a Wikipedia article is easy, and is one of the last steps that students do. The challenging part is to identify which articles or more interesting to males vs. females in a systematic and reproducible way. The obvious thing to do is to get this data from Wikipedia somehow, except that this is hard. Most Wikipedia contributors do not supply their gender, so there may be a strong “self-selection” bias amongst those who do supply gender info and those who so not.

Instead, students use gender information from MovieLens, which is a free online movie recommendation site, where a dramatically higher fraction of participants indicate their gender, and it seems to be much better balanced. To that end, students download a MovieLens dataset with demographic information. They then write code to read three files delimited flat files, each of which contains data about movies, movie ratings, and user demographics. Students then need to invent an appropriate methodology for storing the data in memory. At our institution, where we use Python, this turns into a great exercise in practicing the use of dictionaries. This step of the project is the core programming part of the assignment.

Once students have the data stored, they produce a list of the MovieLens most "male" and "female" movies. Figuring out how to measure the genderedness of a movie is not clear cut. Should it be based on total numbers, or relative fractions? Should it be based on what movies people chose to watch, or what movies people liked? While some guidelines are given in the assignment, the answer to this question is ultimately unspecified, and is left as part of the creative element that students are asked to take.

After having identified two lists of movies, students measure the length of the English Wikipedia articles associated with each of these movies. This is relatively straightforward to do via a short program distributed with the assignment that queries the Wikipedia API. Students are then asked to submit a short paper (including tables of data or graphs) describing how they approached what they did, and what they learned.

The assignment can be found online, described in more detail, at:

http://www.cs.carleton.edu/faculty/dmusican/talks/wikimovies.html

### 4.3 Biographical information

David Musicant is a professor of computer science at Carleton College. His research focuses on solving machine learning and data mining problems, and collaborative human/computing systems. He is currently working to understand the dynamics of people working with online collaborative communities such as Wikipedia.

## References

[1] Joshua E. Blumenstock. Size matters: word count as a measure of quality on Wikipedia. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 1095-1096, New York, NY, USA, 2008. ACM.

[2] Noam Cohen. Define gender gap? Look up Wikipedia's contributor list. *The New York Times*, January 2011.

[3] Shyong (Tony) K. Lam, Anuradha Uduwage, Zhenhua Dong, Shilad Sen, David R. Musicant, Loren Terveen, and John Riedl. WP:Clubhouse?: An exploration of Wikipedia's gender imbalance. In *Proceedings of the 7th International Symposium on Wikis and Open Collaboration*, WikiSym '11, pages 1-10, New York, NY, USA, 2011. ACM.

[4] Thomas Wöhner and Ralf Peters. Assessing the quality of Wikipedia articles with lifecycle based metrics. In *Proceedings of the 5th International Symposium on Wikis and Open Collaboration*, WikiSym '09, pages 16:1-16:10, New York, NY, USA, 2009. ACM.

# 5 Instructor Created Video
# Dean Stevens and Randy Campbell

## 5.1 Description

This past summer we experimented with going bookless in our CS0 course, Crafting the Client Side Web. It is a general-purpose class serving both majors and non-majors alike. We teach the basics of programming through HTML, CSS, and JavaScript. In the past, we have used books, but with web standards changing quickly, they proved to be out of date soon after they were published. We also tried using web only resources, to find that not only were they out of date but they had incorrect information as well. We decided that we would record our own lectures to provide the most up to date (and hopefully correct) information. We used a tool called Camtasia, which records both the screen and the audio and allows for easy export to a variety of formats, including YouTube. This allowed us to use the flipped model of teaching, so our in class time was spent helping students with assignment.

The population at Morningside includes a high percentage of students involved in sports, music, and the fine arts. Thus, on any given day, student absences could be expected. The videos proved to be labor saving by having students watch those videos instead of coming into office hours to get an abbreviated version of the lecture. The videos are ensured that the same material was being presented in both sections of the course. Additionally, our assessment of student work indicates that students understood the material better. Finally, students reported more enjoyment in taking the course in the flipped model rather than the traditional one.

We have found that using these videos allows us to explore class material in a deeper way. We could record videos that explained material not necessary for the course. Students who were interested could then watch those videos on their own time, while those who were not were not forced to sit through a lecture they didn't care about. These videos were so useful that we found ourselves using them in classes outside of the CS0 course. In upper level courses, we could record videos of challenging material so that students could watch the videos on their own time to work out the material at their own pace.

## 5.2 Biographical Information

Dean Stevens has been teaching Computer Science at Morningisde College for thirteen years. While he teaches all courses, he focuses on the applied parts of computer sciences like web technology. He is also the founder of the Video Game Institute for Education and Research (VGIER)

Randy Campbell has been teaching Computer Science at Morningside College for eighteen years. While he is also a Jack-of-all-trades, he focuses on the theoretical side of computer science. His research focuses on Artificial Intelligence like Automated Theorem Proving.

# 6 Supercomputer Project
# Samantha Foley

## 6.1 Computer Architecture Assignment

Computer architecture is often perceived by students as an outdated and boring topic. However, it is a vibrant, diverse and fast paced field of computer science. In order to help students see the diversity and cutting edge of computer architecture, I have developed a final project on supercomputers. Supercomputers are the largest and fastest computers that are designed to tackle grand challenge scientific problems. These machines inherently push the boundaries of computing and often contain the newest technologies and ideas in computer architecture. The project asks groups of students to research one of the top supercomputers in the US as measured by the top 500 list (http://www.top500.org/). Each machine is chosen by the professor to make sure there is a diversity of architectures, funding agencies, and that the information about the machine is easy to obtain. Each group is asked to produce an 8-10 page paper with proper citations, and a 15 minute presentation about their machine. In both the paper and presentation, they must describe the motivation for building the machine, the specific architecture of the machine and highlight a specific scientific application that uses the machine. The bulk of the paper tends to be on the architecture and includes the following aspects: processor design, cache and memory design, storage and file systems, network technology and topology, system and parallel programming software, and performance information including rankings on various lists. The scientific application described must be based on research published in a peer-reviewed venue. It shows students how all of the topics covered during the course of the semester come together to form a single machine, and how many different ways a large-scale system can be built.

This project is used as a final project in the Computer Architecture (CS370) class, which is required for all majors.

This project is nifty in several ways. It provides students with an opportunity to develop “soft skills” (group work, writing, oral presentation), conduct research into current cutting-edge technology relevant to the class, and see how computer science can be applied to a wide range of fields. These are all skills that will not only help them prepare for careers in any computer science field, but also deepen and reinforce the fundamental computer architecture topics covered in the class.

So often we only have time to present the very basics of a sub-area of computer science before needing to move on to another topic. This leaves us presenting technology as it existed decades ago in order for it to be digestible by the students. Current technology tends to be vastly more complex, but mostly variations on the same themes. By presenting the traditional computer architecture curriculum throughout the semester, and then spending a week for students to present their supercomputers, students get to see a single new architecture in depth, and several others through their classmates’ presentations, thus everyone can see the breadth of architectures and begin to identify the themes among them.

Computing plays an ever-increasing role in other disciplines and this is also something that is not often discussed in the interest of time during classes. Using the project to have students investigate one application in-depth, and then share it, students are be able to see a wide range of scientific applications using supercomputers.

Full assignment description and links given to students:
http://cs.uwlax.edu/~ssfoley/cs370.html

## 6.2 Biographical Information

Samantha Foley is an Assistant Professor in the computer science department at University of Wisconsin-La Crosse. Her research and teaching interests are in computer architecture, parallel and distributed computing, and scientific computing, specifically how to make parallel and distributed computing easier to learn and use for students and scientists. She earned her PhD in computer science from Indiana University in 2010, and spent the next two years at Oak Ridge National Laboratory.