AC 2010-1153: TEACHING INTERACTIVELY WITH GOOGLE DOCS

Edward Gehringer, North Carolina State University

Ed Gehringer, efg@ncsu.edu, is Associate Professor of Computer Science and Computer Engineering at North Carolina State University. His main research area is collaborative learning technology. He received his Ph.D. degree from Purdue University, and taught at Carnegie Mellon University, and Monash University in Australia.

© American Society for Engineering Education, 2010

Teaching Interactively with Google Docs

Abstract

Active learning has repeatedly been shown to be important to retention of what is taught in class. An important challenge is devising enough active-learning exercises to reinforce each important concept and prevent students from "tuning out" after the first fifteen minutes of class. Since many students nowadays carry laptops, we should look for ways to use them in active learning—rather than let them be a distraction during class. Using Google forms, students can give feedback on their laptops during class. Google docs can be used in a myriad of ways, such as for collecting answers to questions, submitting computer code, filling out tables and tableaus, or reporting on Web research done during class. It is quite easy to use Google docs in almost any class, providing the kind of interactivity that is missing from a traditional lecture.

1. Introduction

It has been about 10 years since universities began to require students to own laptops. It is safe to say that the anticipated educational benefits have not accrued. Indeed, some recent studies [1, 2] have reported that students who bring laptops to class actually learn *less* than those who do not. However, banning laptops is problematical [4] because it forecloses such positive uses of laptops as taking notes and working problems in class. This has led many educators to realize, belatedly, that "lecturing is dead" [5]. To retain students' attention, classes must become more interactive. Laptops can fill this bill.

In recent years, software applications such as DyKnow [6], Ubiquitous Presenter [7], MessageGrid [8], and Group Scribbles [9] have been developed to bring true interactivity to the laptop classroom. They provide tremendous help in conveying complex concepts. But they have *not* caught on with the vast majority of engineering faculty. They require an infrastructure that places demands on the professor to learn how to use these applications, and on the IT staff to install and support them. In years to come, they will penetrate further, as they should. But for most faculty, they are not an instant solution.

An attractive alternative is Google docs. One can typically learn how to create Google docs in less than an hour. No infrastructure is needed, except for a wireless network. Google supplies the servers and the storage. Google docs are versatile enough to be used in almost any engineering class. The remainder of this paper will show how to get started, and provide a sampling of the many uses that Google docs can play in the wireless classroom.

2. Creating and Using Google Docs

It is very easy to create a Google doc. First of all, one needs a Google account. Your gmail account will work, if you have one. If you do not, you need to be invited to open one by a current gmail user.

Copyright © 2010, American Society for Engineering Education

Once you have an account, log in at http://docs.google.com, and you will be presented with a screen that looks something like this:

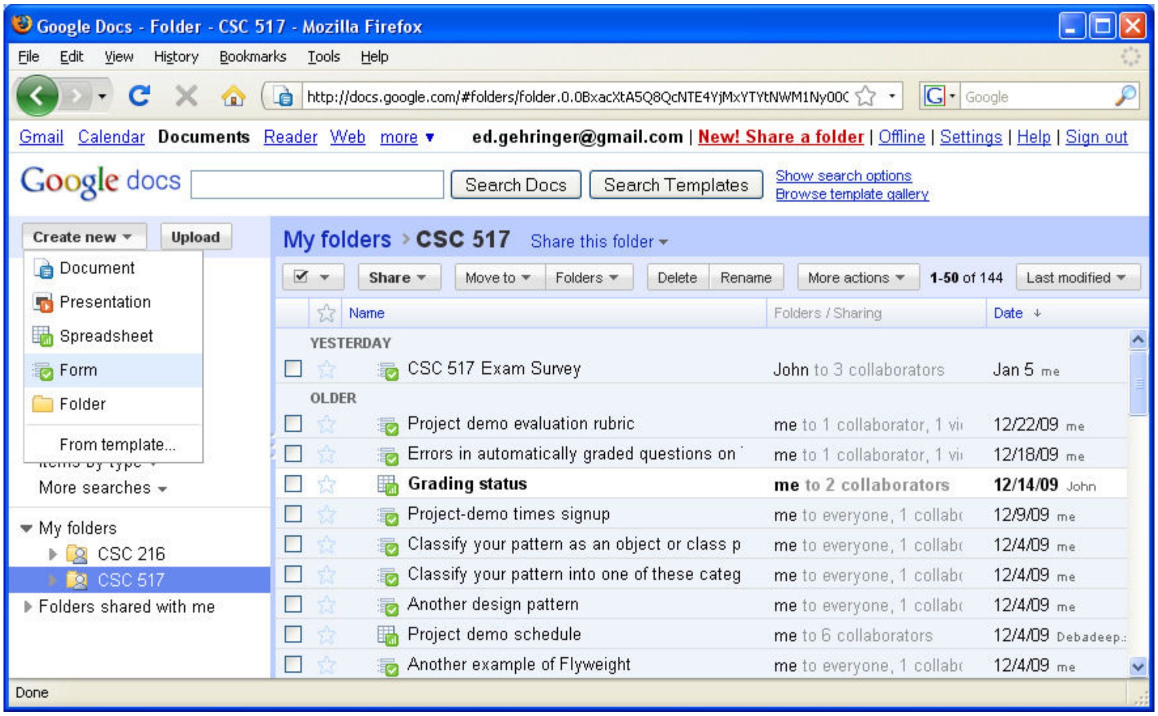


Figure 1. Creating a Google doc

Click on the "Create new" button and select the kind of doc you want to create. In this paper, we will be discussing Forms and Spreadsheets. The illustration above is of creating a Form.

When you create a form, you are presented with a boilerplate form that contains two questions. You can change the title or type of each question, as shown at the top of the next page. You can add questions, as shown in Figure 3. If you have several questions to add, it is easiest to add them first, then edit the title, type, and help text of each new question.

The students, of course, need to have access to the form. The URL of the form can be posted on the class Web site, or placed in online lecture notes. The URL should be on a page that is protected from unauthorized access, or on a page with a <NOINDEX> tag to prevent it from being indexed by search engines. This reduces the possibility that spammers, or bots out in cyberspace, will submit gibberish and thereby pollute the student answers.

Copyright © 2010, American Society for Engineering Education

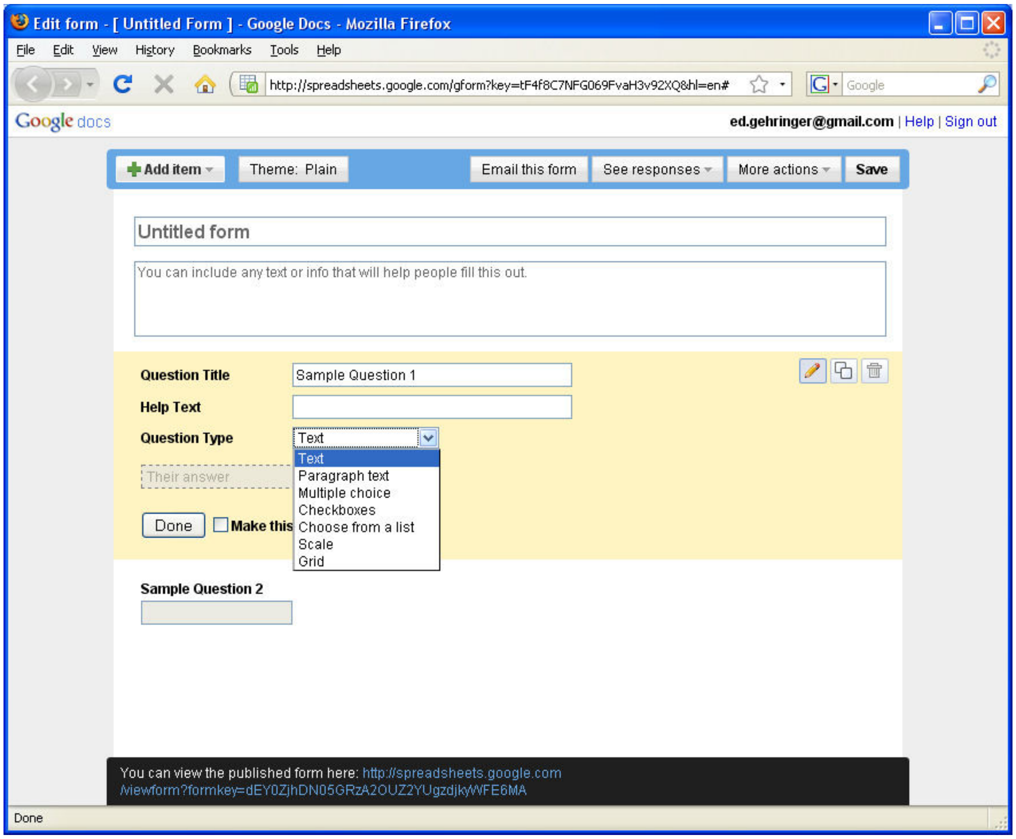


Figure 2. Editable boilerplate for a new Form

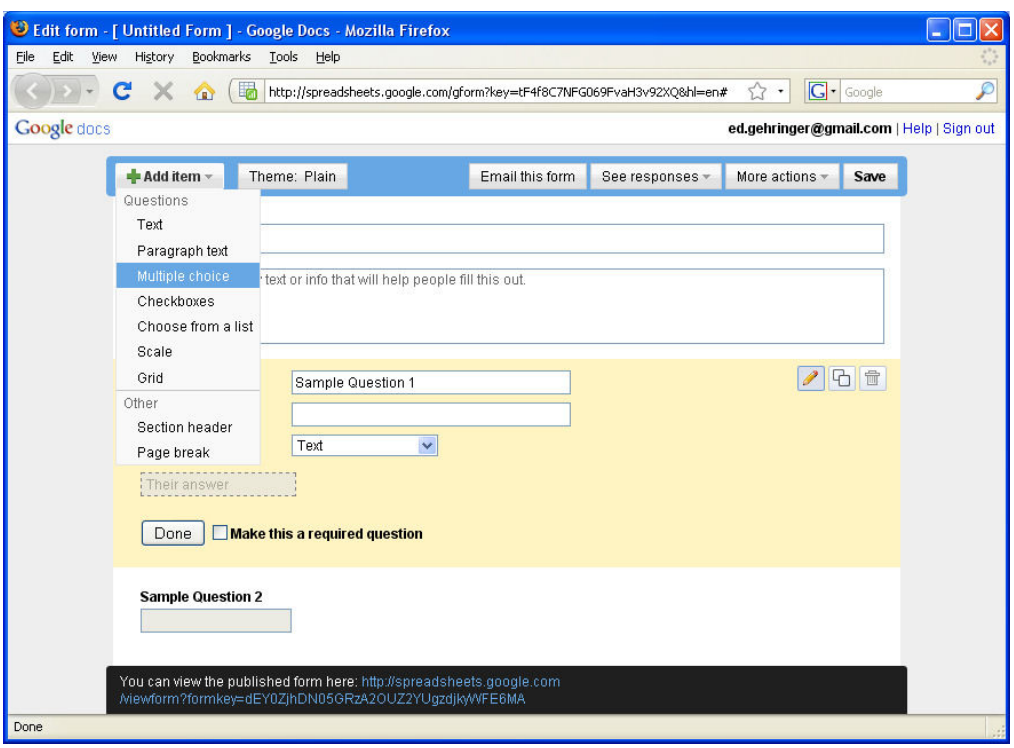


Figure 3. Adding a question to a form

Copyright © 2010, American Society for Engineering Education

Page 15.1177.4

Creating a spreadsheet is even easier … just select "Spreadsheet" from the dropdown shown in Figure 1. One additional step is necessary for a spreadsheet that you are going to use for an exercise: you need to make it editable by any student in your class (Figure 4). You do that by going to the Share menu and selecting, "Get the link to share". You will be required to save the document first. Then you need to check the two boxes to allow others to view the document without logging in, and to edit the document (Figure 5).

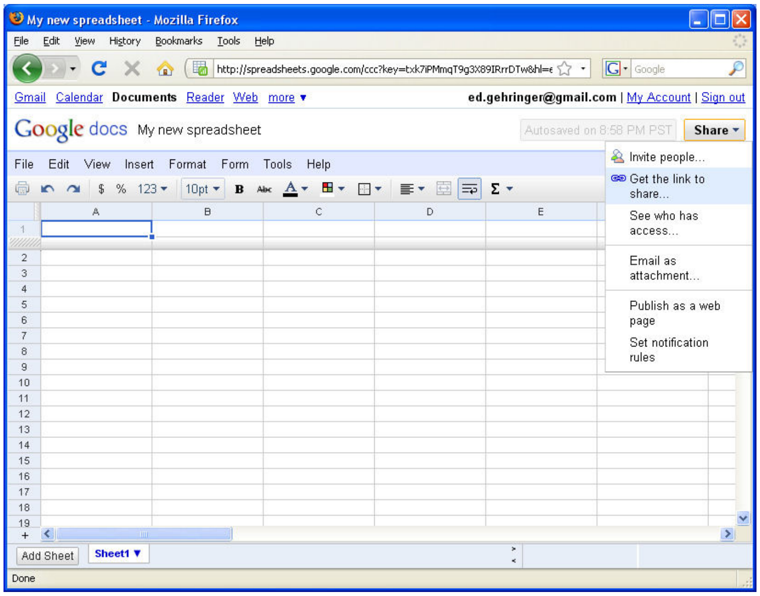


Figure 4. Sharing a document

Because I only want students in the class to be able to edit the document, I place the link to my documents on a page that the students need to log in to view. There are other ways of achieving the same effect, such as by sharing the document with every student in the class, but that would have been more trouble for me.

Forms can be shared with students in a similar fashion. In this case, you go to the bottom of the form, where it says, "You can view the published form here:" and clicks on the link to the right. This is the link that you give students so that they can use the form.

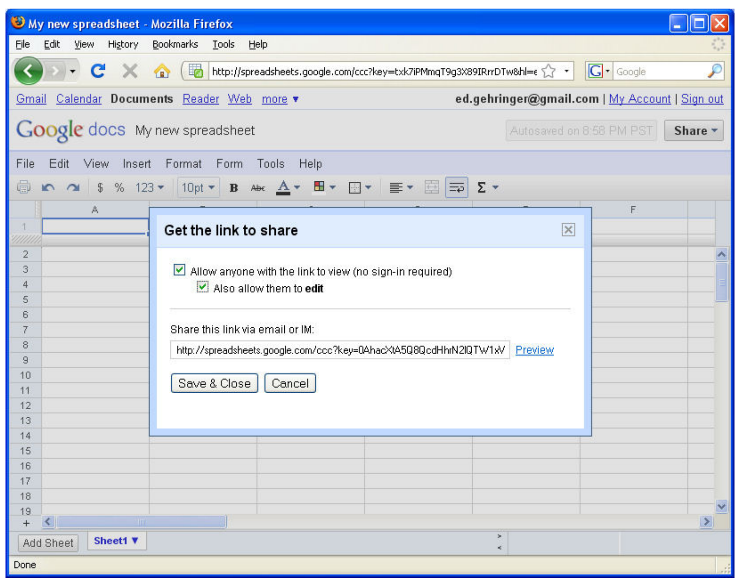


Figure 5. Making a spreadsheet editable by any student

3. Uses of Google Forms

While spreadsheets are useful in their own right (Section 4), most classroom uses of Google docs involve Forms. The most basic usage is using a form to collect answers to questions posed in class. In a traditional lecture, the instructor would call on one student to answer a question. The rest of the class could sit passively, while the selected student occupies the hot seat. With a form, everyone can answer (well, everyone with a laptop—since there will be some students without laptops, the instructor can encourage them to pair with those who have

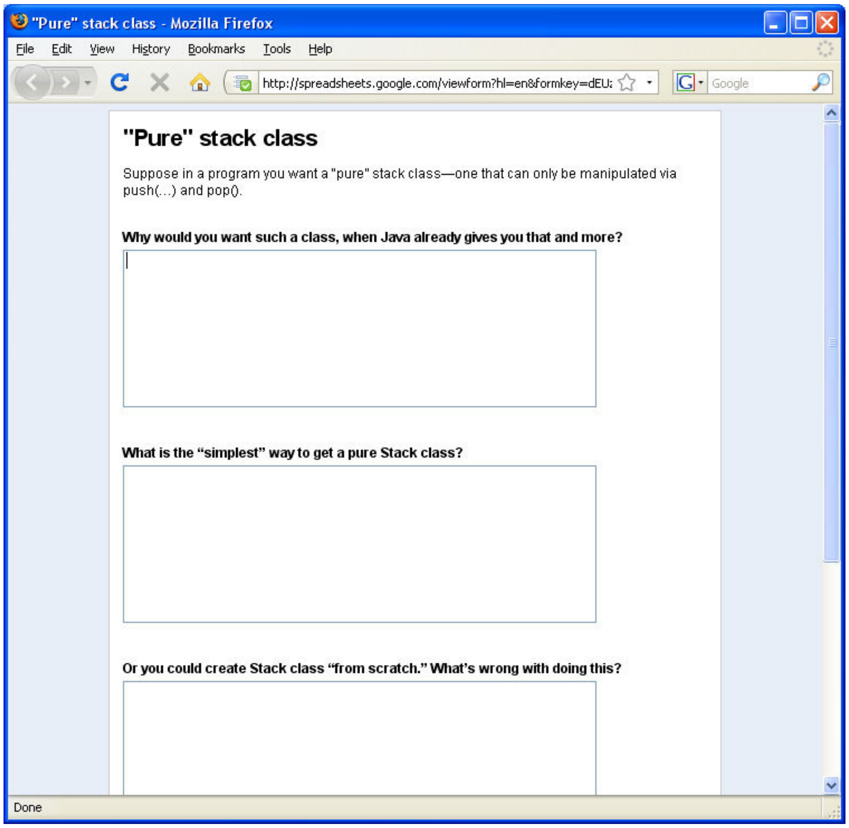
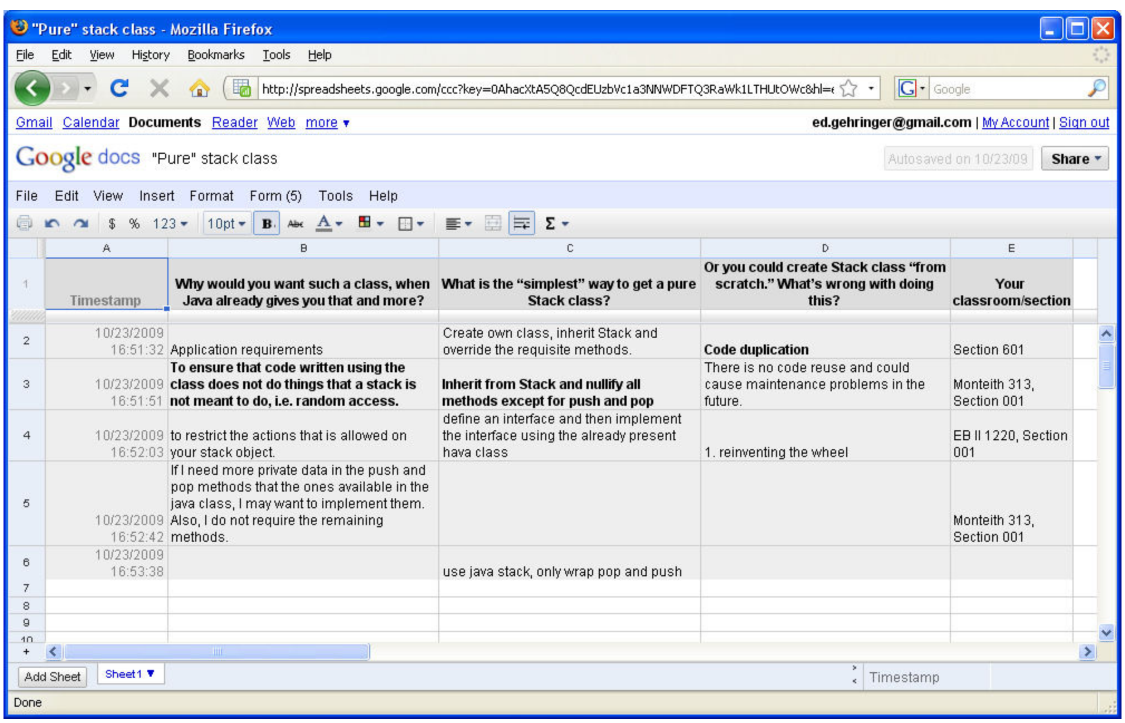


Figure 6. Collecting answers to prose questions

laptops and discuss their answer before submitting it). The instructor can **collect prose answers**, as shown in Figure 6. Or, the questions might have multiple-choice answers. As students submit the form, their answers populate rows of an associated spreadsheet (Figure 7). In that figure, I've used boldface to highlight the answers that I wanted to discuss in class.



	A	B	C	D	E
1	Timestamp	**Why would you want such a class, when Java already gives you that and more?**	**What is the "simplest" way to get a pure Stack class?**	**Or you could create Stack class "from scratch." What's wrong with doing this?**	**Your classroom/section**
2	10/23/2009 16:51:32	Application requirements	Create own class, inherit Stack and override the requisite methods.	**Code duplication**	Section 601
3	10/23/2009 16:51:51	**To ensure that code written using the class does not do things that a stack is not meant to do, i.e. random access.**	**Inherit from Stack and nullify all methods except for push and pop**	There is no code reuse and could cause maintenance problems in the future.	Monteith 313, Section 001
4	10/23/2009 16:52:03	to restrict the actions that is allowed on your stack object.	define an interface and then implement the interface using the already present hava class	1. reinventing the wheel	EB II 1220, Section 001
5	10/23/2009 16:52:42	If I need more private data in the push and pop methods that the ones available in the java class, I may want to implement them. Also, I do not require the remaining methods.			Monteith 313, Section 001
6	10/23/2009 16:53:38		use java stack, only wrap pop and push		

Figure 7. Spreadsheet populated as a form is submitted

Copyright © 2010, American Society for Engineering Education

Forms can be used to **submit computer code**, as illustrated in Figures 8 and 9. The instructor provides a partial solution to save time, leaving out key portions that the students are responsible for filling in. After the students submit their answers, the instructor can run them to see if they work.

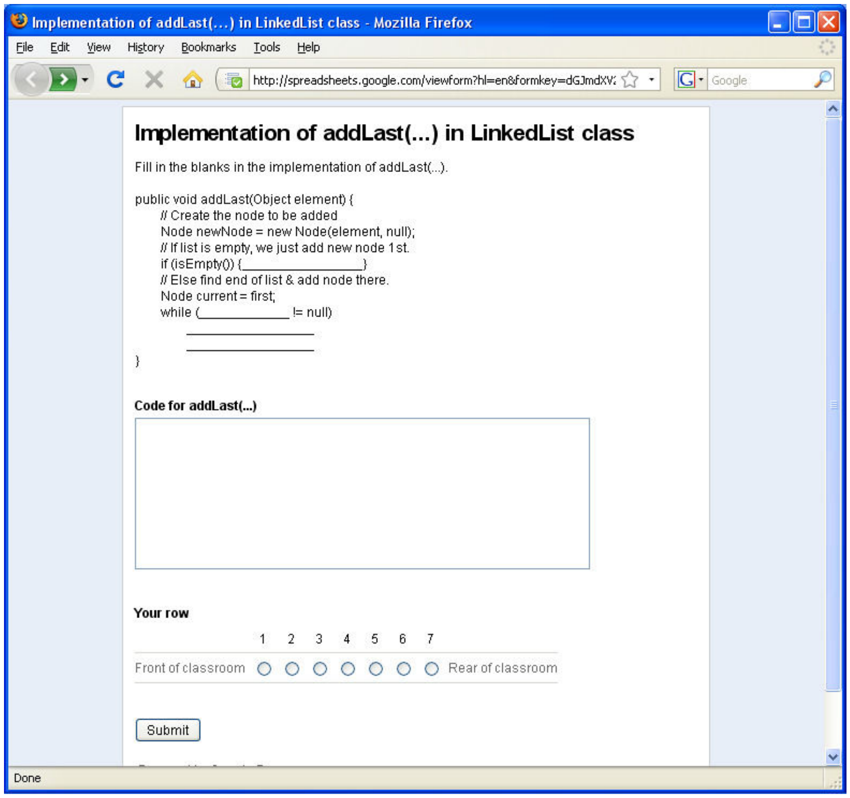


Figure 8. Form for having the students write code

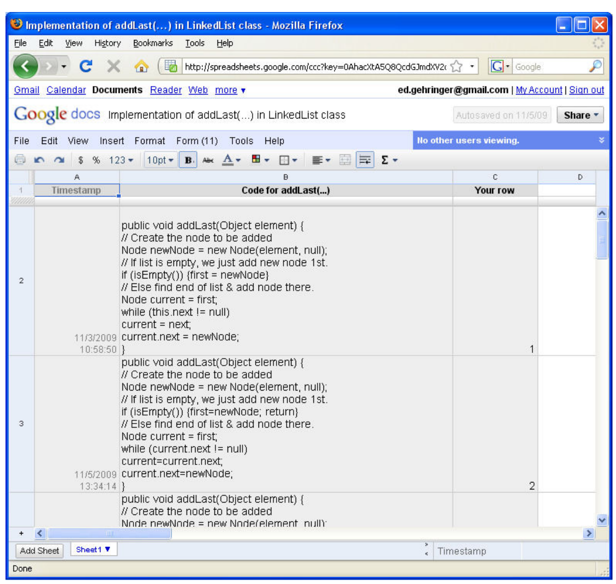


Figure 9. Spreadsheet in which code appears

Copyright © 2010, American Society for Engineering Education

A related use is having the students **submit answers to an in-class exercise**. Figure 10 shows code fragments that needed to be filled in to achieve a desired effect in a linked-list simulator. (The names of the students have been changed to preserve their privacy.)

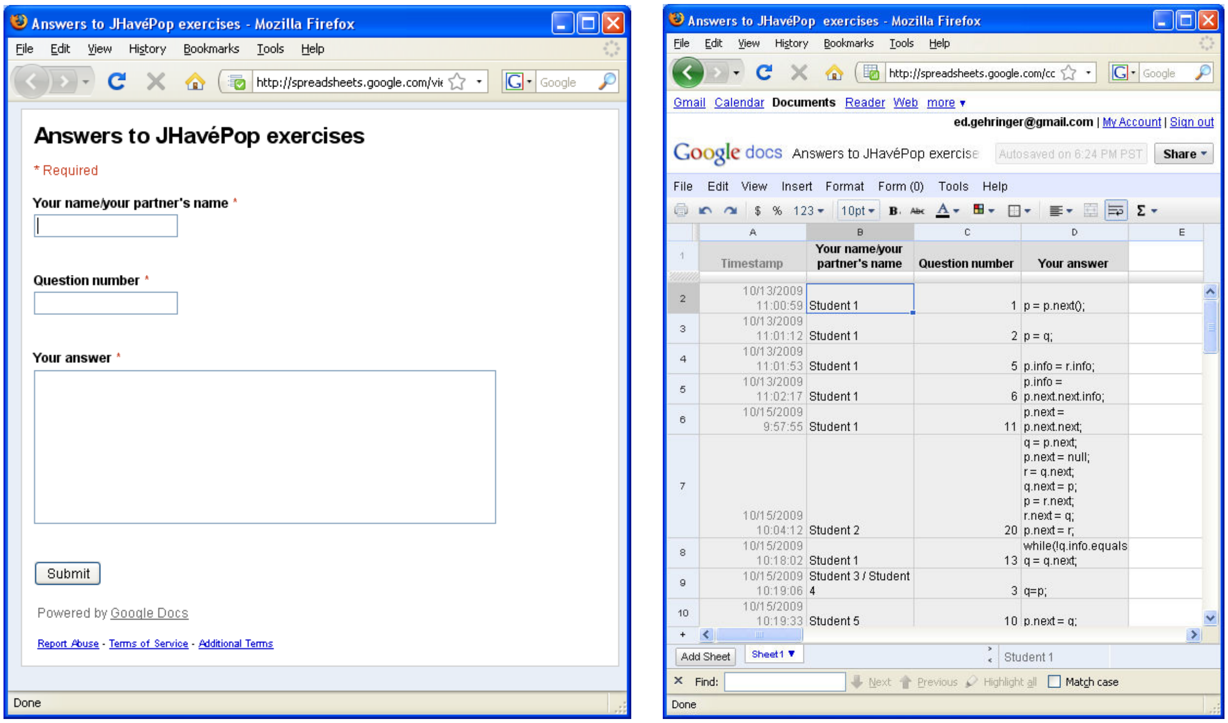


Figure 10. Using a form to collect answers to in-class exercises

When a student is called to the board to work a problem in a traditional classroom, (s)he is often asked to **show the steps in a derivation**. The same effect can be achieved with a form, where successive questions ask what happens during successive steps. Figure 11 shows a form for collecting this information, and Figure 12 presents the associated spreadsheet.

Note that the last question asks students which row of the classroom they are sitting in. I use this question in my classes to spot areas of the classroom where students are not engaged, and also to facilitate competitions among rows to answer questions successfully.

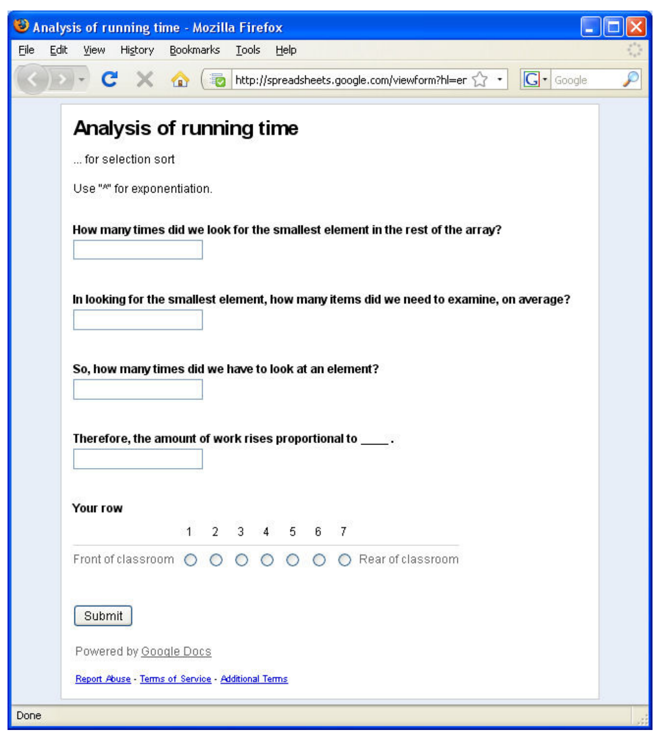
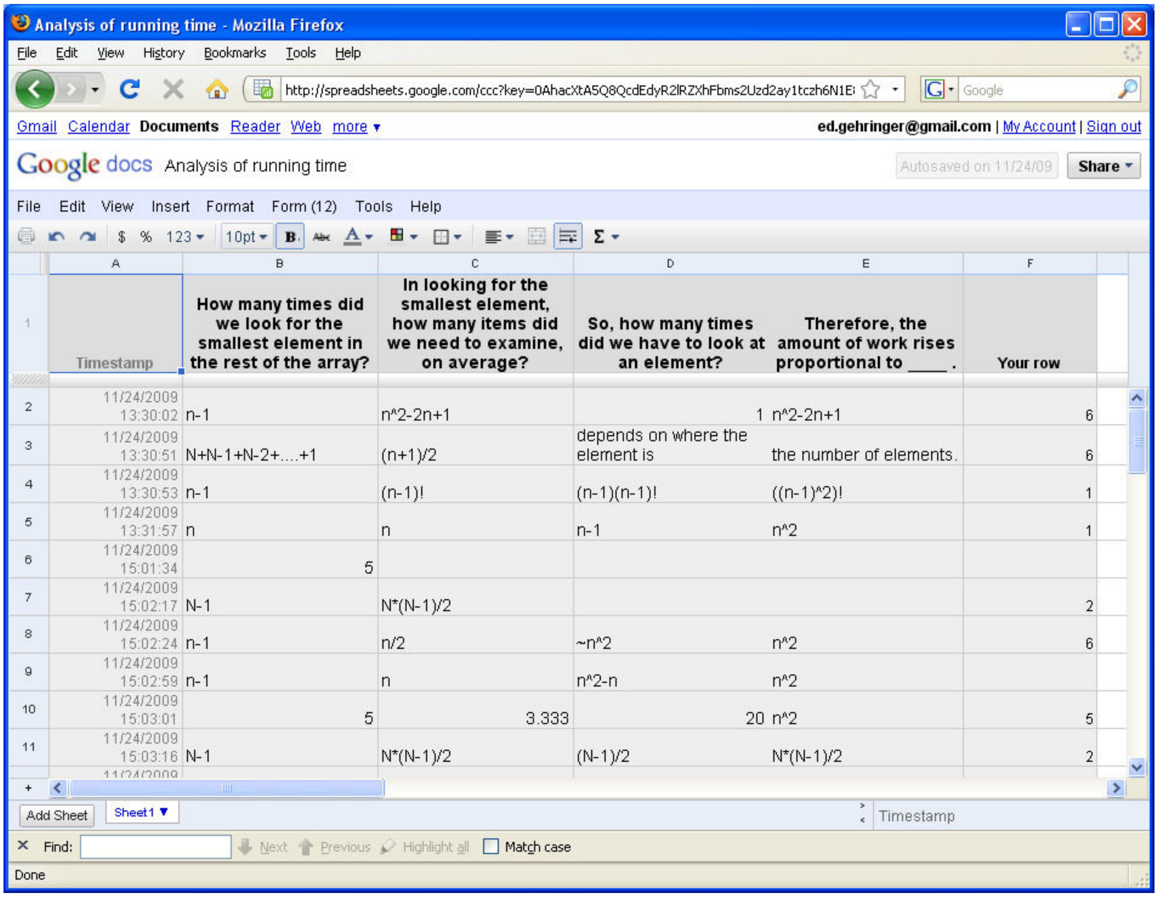


Figure 11. Asking for the steps in a derivation

Copyright © 2010, American Society for Engineering Education



	A	B	C	D	E	F
1	Timestamp	How many times did we look for the smallest element in the rest of the array?	In looking for the smallest element, how many items did we need to examine, on average?	So, how many times did we have to look at an element?	Therefore, the amount of work rises proportional to ____ .	Your row
2	11/24/2009 13:30:02	n-1	n^2-2n+1	1	n^2-2n+1	6
3	11/24/2009 13:30:51	N+N-1+N-2+....+1	(n+1)/2	depends on where the element is	the number of elements.	6
4	11/24/2009 13:30:53	n-1	(n-1)!	(n-1)(n-1)!	((n-1)^2)!	1
5	11/24/2009 13:31:57	n	n	n-1	n^2	1
6	11/24/2009 15:01:34	5				
7	11/24/2009 15:02:17	N-1	N*(N-1)/2			2
8	11/24/2009 15:02:24	n-1	n/2	~n^2	n^2	6
9	11/24/2009 15:02:59	n-1	n	n^2-n	n^2	
10	11/24/2009 15:03:01	5	3.333	20	n^2	5
11	11/24/2009 15:03:16	N-1	N*(N-1)/2	(N-1)/2	N*(N-1)/2	2

Figure 12. Spreadsheet showing student answers for different steps in a derivation

One use that shows the power of this technique is to assign students in different areas of the classroom different questions to answer. They submit their answers, identifying the specific question they are answering. The resulting spreadsheet can then be sorted to **produce a table**. Figure 13 shows how different groups of students were assigned to run a sorting simulator with different parameters, and the results were compiled to form a table that shows how run time varies depending on sorting algorithm, initial configuration, and size of data set. Figure 14 shows how different students were engaged to classify different software design patterns along various dimensions, again producing a table.

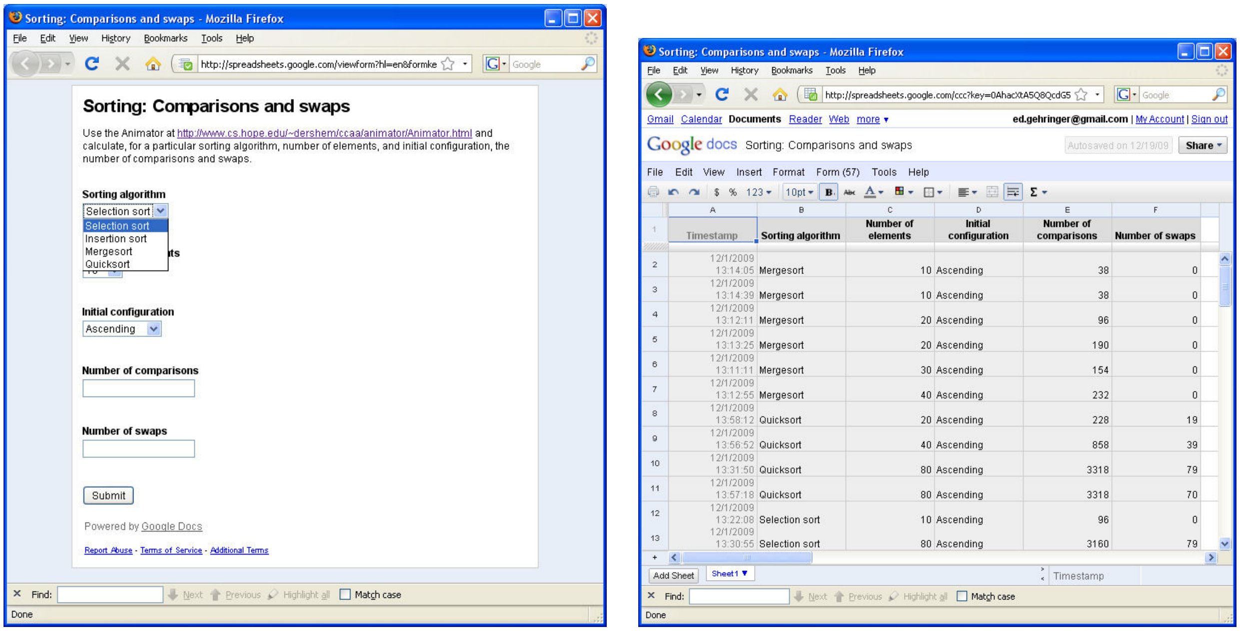


Figure 13. Working cooperatively with a sort simulator to produce a table of sorting times

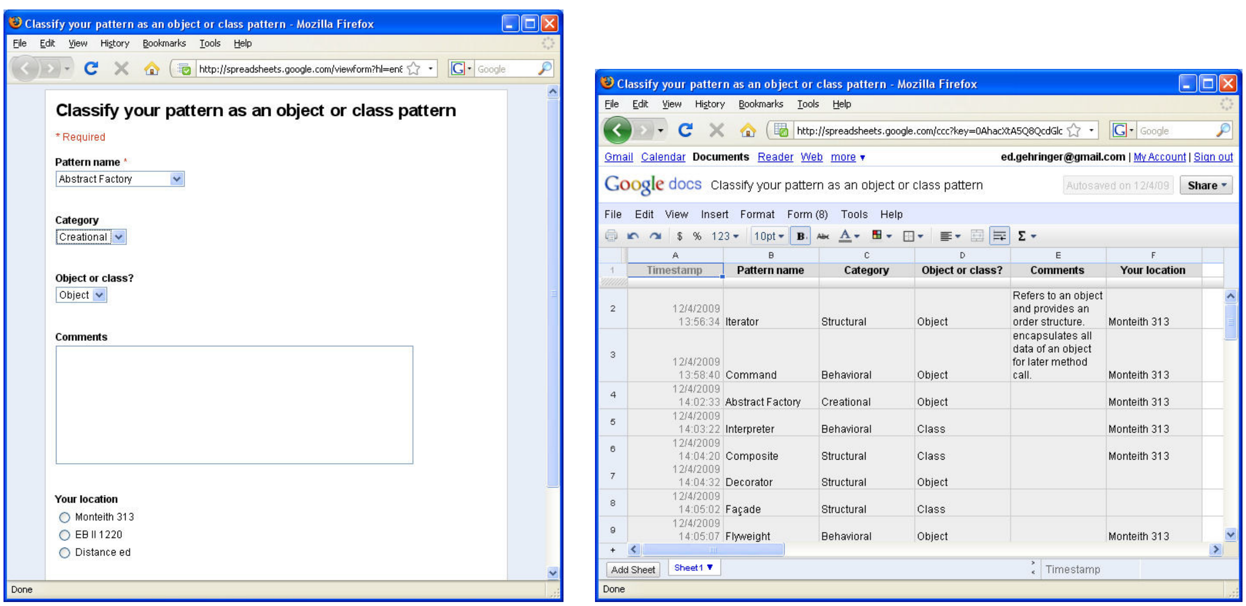
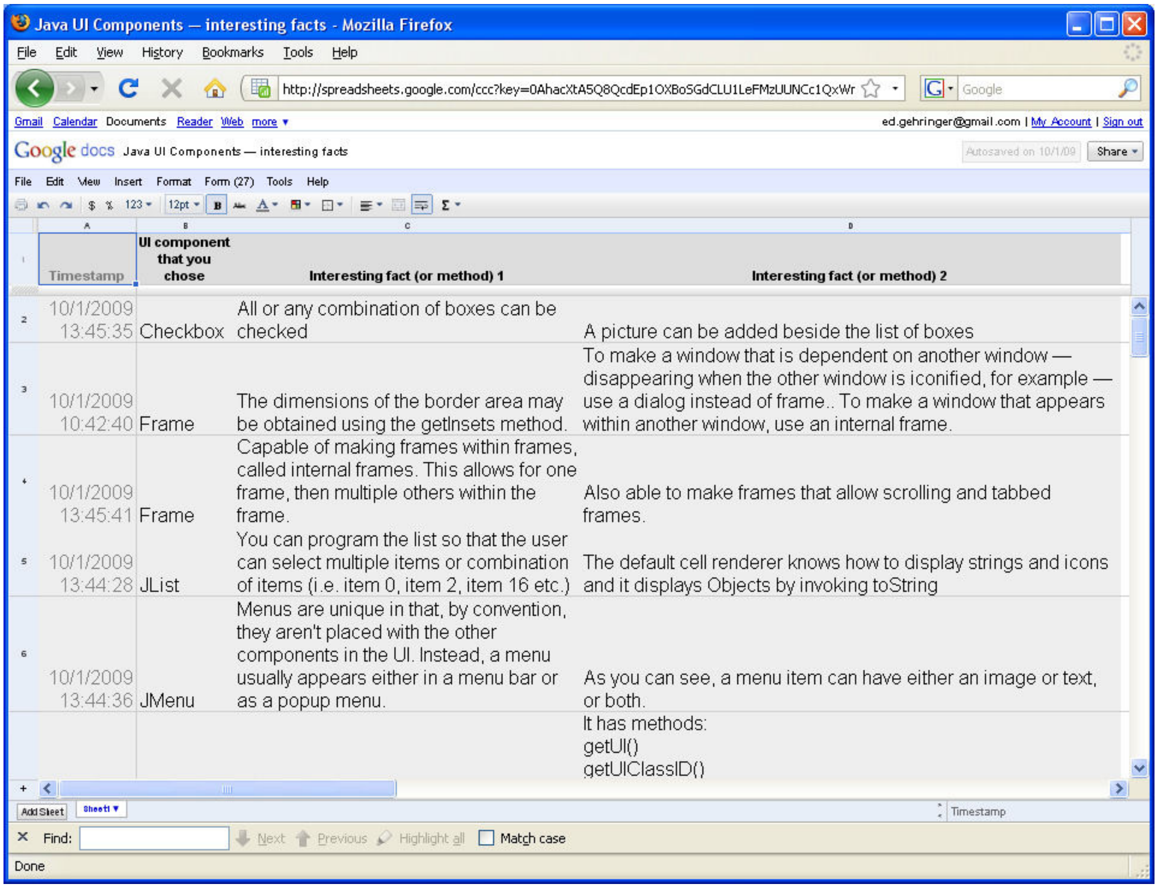


Figure 14. Cooperatively classifying software design patterns

Using the wireless network, students can be asked to look up material on the Web during class. Again, they can be asked to find examples of different artifacts or phenomena on the Web, and present them to the class. This is sometimes called a "**scavenger hunt**." The instructor will have the opportunity to see the examples as they are coming in, and choose the more interesting ones for class presentation and instruction. Figure 15 shows different user-interface components that a second-semester programming class found and presented to their classmates.

Copyright © 2010, American Society for Engineering Education



Timestamp	UI component that you chose	Interesting fact (or method) 1	Interesting fact (or method) 2
10/1/2009 13:45:35	Checkbox	All or any combination of boxes can be checked	A picture can be added beside the list of boxes
10/1/2009 10:42:40	Frame	The dimensions of the border area may be obtained using the getInsets method.	To make a window that is dependent on another window — disappearing when the other window is iconified, for example — use a dialog instead of frame.. To make a window that appears within another window, use an internal frame.
10/1/2009 13:45:41	Frame	Capable of making frames within frames, called internal frames. This allows for one frame, then multiple others within the frame.	Also able to make frames that allow scrolling and tabbed frames.
10/1/2009 13:44:28	JList	You can program the list so that the user can select multiple items or combination of items (i.e. item 0, item 2, item 16 etc.)	The default cell renderer knows how to display strings and icons and it displays Objects by invoking toString
10/1/2009 13:44:36	JMenu	Menus are unique in that, by convention, they aren't placed with the other components in the UI. Instead, a menu usually appears either in a menu bar or as a popup menu.	As you can see, a menu item can have either an image or text, or both.
			It has methods: getUI() getUIClassID()

Figure 15. Some of the interesting facts uncovered during a Web "scavenger hunt"

One very useful application is to have the class **create a grading rubric** that will be used to assess their own homework. This amounts to a two-step exercise in creating tables, as described below. It is useful because it forces the students to think more reflectively about the assignment they have been given, and it also helps them to "buy into" the rubric that is ultimately developed, which lends more credibility to the grading.

In Step 1, the instructor identifies some broad dimensions that should be considered in assessment. The students are asked to describe what should be covered in each dimension. They are also encouraged (via the last question, not shown) to suggest other areas for inclusion in the rubric. The instructor looks over the student responses after class, and devises a wording for each dimension. During the next class (Step 2), the students are presented with these wordings, and asked to describe the characteristics of a program that is exemplary, proficient, adequate, etc. in each of these areas. Their responses can be combined to produce an extensive set of rubric advice, which tells the assessor (instructor, TA, etc.) how to grade a homework submission on each of these dimensions. Forms for both steps are shown in Figure 16.

Copyright © 2010, American Society for Engineering Education

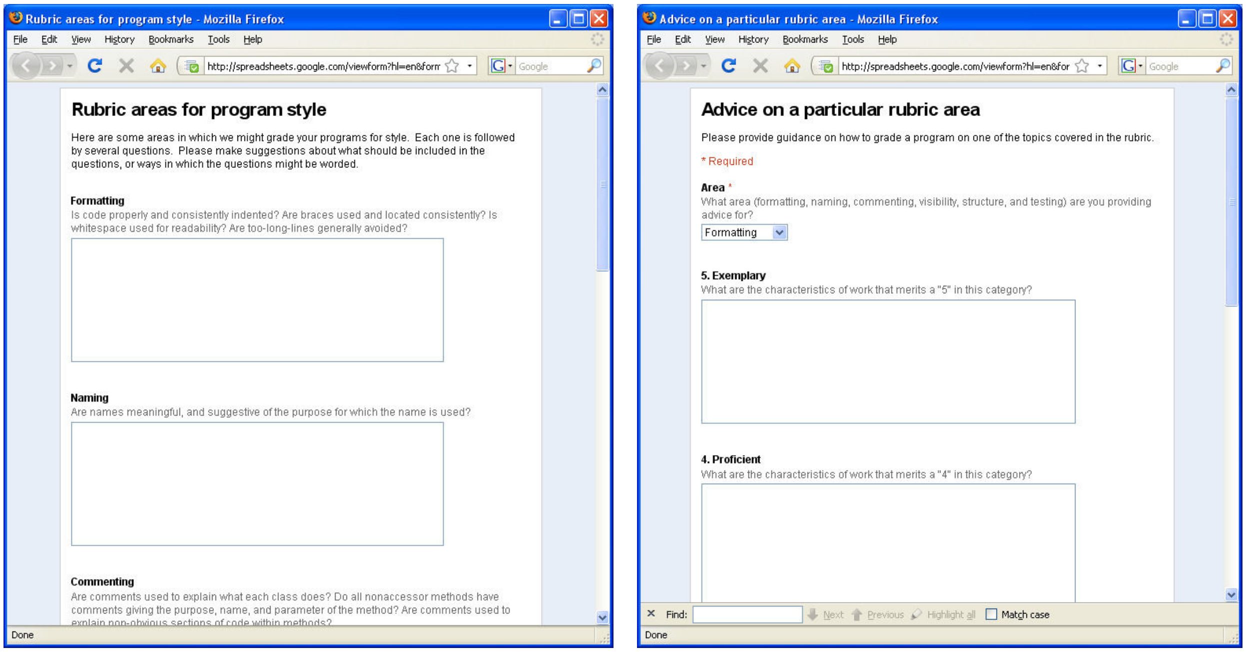


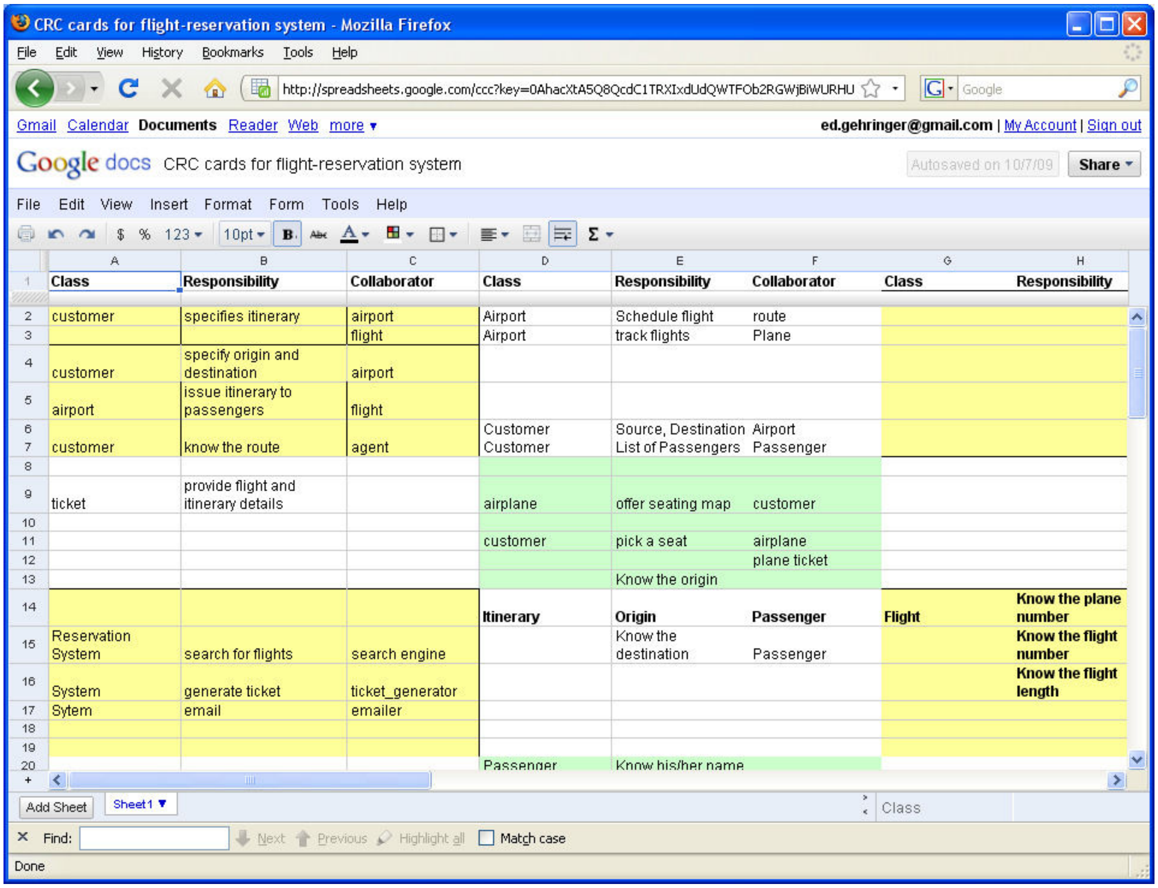
Figure 16. The two-step process of producing a rubric using student input

The last use is to **take polls** in class, in much the same way as is done with clickers. Students are asked to answer a multiple-choice question (or questions), and the spreadsheet tabulates the number of students choosing each response. This information can be presented to the instructor (and to the class) graphically using Google charts. The paper contains no screenshot of this use, because it is under development, and should be available by the end of the Spring 2010 semester.

4. Uses of Google Spreadsheets

While most classroom exercises lend themselves to Google forms that produce spreadsheets, it is sometimes helpful to let students use the spreadsheets directly. One case is where the students need to fill out a tableau. This would be awkward with forms, because questions always appear sequentially; there is no way to make multiple blanks appear in the same row. Tableaus correspond to the Angelo-Cross [10] Memory Matrix assessment technique. The instructor creates a Google spreadsheet that has different colored rectangles, where each rectangle contains a set of cells. The spreadsheet is shared with the students, who write directly on one of the colored rectangles, and fill in the individual cells in the rectangle. This can be used, for example, to identify different properties of a set of chemical compounds, or to fill in CRC (class-responsibility-collaborator) cards in object-oriented software design. Figure 17 shows the completed CRC cards.

Copyright © 2010, American Society for Engineering Education



	A	B	C	D	E	F	G	H
1	Class	Responsibility	Collaborator	Class	Responsibility	Collaborator	Class	Responsibility
2	customer	specifies itinerary	airport	Airport	Schedule flight	route		
3			flight	Airport	track flights	Plane		
4	customer	specify origin and destination	airport					
5	airport	issue itinerary to passengers	flight					
6				Customer	Source, Destination	Airport		
7	customer	know the route	agent	Customer	List of Passengers	Passenger		
8								
9	ticket	provide flight and itinerary details		airplane	offer seating map	customer		
10								
11				customer	pick a seat	airplane		
12						plane ticket		
13					Know the origin			
14				Itinerary	Origin	Passenger	Flight	Know the plane number
15	Reservation System	search for flights	search engine		Know the destination	Passenger		Know the flight number
16	System	generate ticket	ticket_generator					Know the flight length
17	Sytem	email	emailer					
18								
19								
20				Passenger	Know his/her name			

Table 17. Tableau for creating CRC cards

I've also used this approach to develop rubrics, placing the questions down the left side of the tableau and the advice across each row. One hazard of this approach is that immature students can "scribble" all over the spreadsheet. In my sophomore class, one student pasted the same irrelevant expression into dozens of cells all over the spreadsheet. This was not a problem in my graduate class, however.

5. Uses of Google forms for administration

Thus far, we have discussed using Google forms for in-class exercises. If Google docs are used for this purpose, it also makes sense to use them outside of class. It's my practice to invite students to create a class roster at the beginning of each semester. They have the option of sharing their hometown, schools that they've attended, unusual facts about themselves, and their Web pages. They can create the roster by filling out a Google form, as shown in Figure 18.

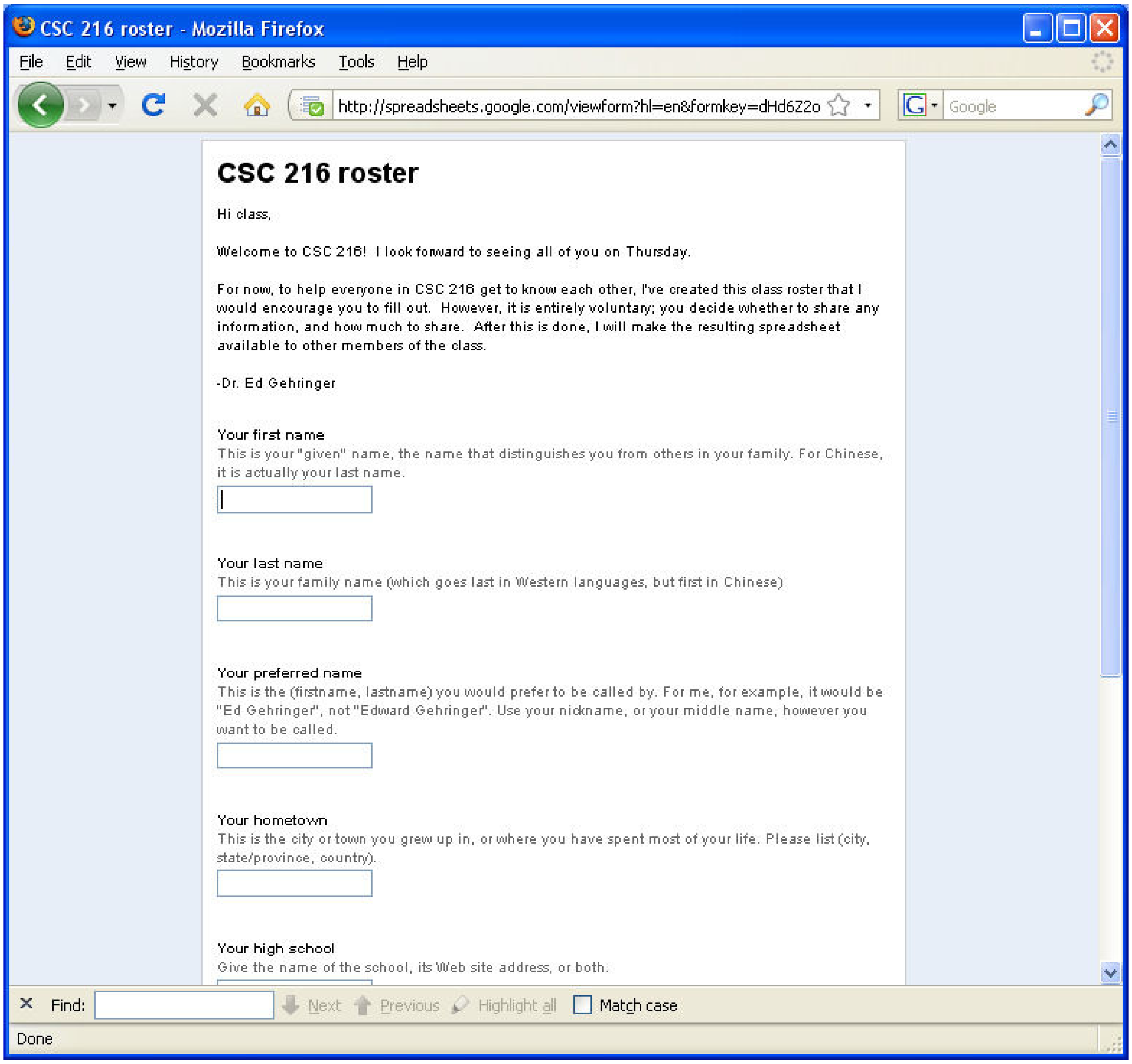


Figure 18. Class roster

Other administrative uses include registering (self-selected) partnerships for doing particular homework assignments, and registering for specific project topics when students are asked to choose from a list of topics for their project. Forms are helpful, but not ideal, for these uses. It's possible for student *A* to register a partnership with student *B*, while student *B* says (s)he's partnering with student *C*, for example. These discrepancies need to be resolved manually. In signing up, there's no way to enforce a limit on how many students (or teams) can sign up for the same topic. I tried using a Google spreadsheet, identifying particular cells as the "slots" for a particular topic. But some students overwrote each other's entries, or placed their names in cells adjacent to the specified cells, because the specified cells were already taken.

Copyright © 2010, American Society for Engineering Education

6. Student reaction

Students liked using forms to submit their work; they felt that they got more experience working through examples than they would otherwise have obtained. The most common dislike was other students "spamming" the spreadsheet, when spreadsheets were used directly, instead of forms. One persistent problem was that some students did the exercises, while others didn't. Sometimes whole areas of the classroom appeared "tuned out." It should help to have students submit their names, and then grade the exercises. But in a large class with several exercises per class period, this would be a lot of work. Grading could be mostly automated for multiple-choice questions, but not for questions that have freeform answers. Some students felt they didn't have enough time to do the exercises, and asked for them to be posted well before class. But this could backfire, if students worked the exercises in advance and then spent class time reading e-mail.

Some students also requested that they be allowed to view the spreadsheets that the class had created by filling in forms. I sometimes allowed this, but there are caveats. First, one would not want to allow students to see others' answers to multiple-choice questions during the class, as it would bias their selections. Sharing answers after the class is also problematical, because some of those answers will be wrong. It would require considerable work to eliminate the wrong answers from the spreadsheet. In a case where students are simply reporting "interesting facts," or creating examples, it might be reasonable to share all of these with the class. Otherwise, caution would be warranted.

7. In-class procedure

Many classrooms nowadays are equipped with computers. When using Google forms, it's helpful to be able to look at two screens—the classroom computer and a laptop. The classroom computer can show the instructions for the exercise, or background material. The laptop can be used to observe the responses coming in. This allows one to perform the following tasks:

- Adjust the column boundaries and font sizes for better visibility when the answers are displayed to students.
- Remove impertinent responses submitted by students.
- Sort the responses, if a table is being constructed.
- Highlight interesting answers (e.g., by boldfacing or increasing the font size) for later discussion.

8. Future development

In Spring 2010, an independent study is underway to provide a way to visualize the answers to multiple-choice questions using Google charts. For example, a pie chart could be displayed showing how many students chose each answer to a particular question. This will allow Google

Copyright © 2010, American Society for Engineering Education

forms to accomplish many of the same tasks that are commonly performed using clickers, without requiring the students to bring an extra device to class.

Another enhancement is support for in-class contests between different groups of students, such as the occupants of different rows in the classroom. The instructor could identify the correct answer by setting the value of a spreadsheet cell, and the spreadsheet could calculate the number of students in each group who had given the correct answer.

9. Summary

Google forms and spreadsheets can be used in class to obtain responses from students that can immediately be displayed to their peers. It is a more effective use of class time than asking questions to individual students, because with Google docs, all students have the opportunity to answer. If students without laptops pair with students who have brought laptops, the think-pair-share paradigm of active learning is directly supported. It is more effective than having students come to the board to write answers because (1) the instructor gets to pre-screen the responses to select ones that are interesting, and (2) it eliminates the downtime while a student is writing an answer on the board.

We have identified several ways that Google docs can be used. Google forms can be used during class to—

- collect answers to questions, either prose or multiple choice;
- submit computer code, which can then be run by the instructor;
- submit answers to exercises worked in class;
- show steps in a derivation;
- produce a table, by having different students fill in different rows of the table;
- present results of a Web “scavenger hunt” to share with the class;
- create a grading rubric for a homework assignment in the class; and
- take polls in a class, in much the same way that clickers do.

Outside of class, Google forms can be used to develop a class roster, or register self-selected partnerships for a group project.

Google spreadsheets can be used directly to fill out a tableau, thus implementing the Memory Matrix classroom assessment technique.

This set of uses was developed in a single semester by a single instructor. Obviously it does not represent a comprehensive list. The author invites you, the reader, to add to this list by filling out the form at http://tinyurl.com/gdoc-use-describe. Responses of others will be viewable at http://tinyurl.com/gdoc-use-view.

Bibliography

[1] Fried, Carrie B. “In-class laptop use and its effects on student learning.” *Computers & Education* 50:3, April 2008, pp. 906–914.

Copyright © 2010, American Society for Engineering Education

[2] Wurst, C; Smarkola, C.; Gaffney, M.A. "Ubiquitous laptop usage in higher education: Effects on student achievement, student satisfaction, and constructivist measures in honors and traditional classrooms." *Computers & Education* 51:4, December 2008, pp. 1766–1783.

[3] Barak, M.; Lipson, A.; Lerman, S. "Wireless laptops as means for promoting active learning in large lecture halls." *Journal of Research on Technology in Education* 2006, pp. 245–263.

[4] McWilliams, G. "Laptops in classrooms not working out as hoped." *Wall Street Journal*, October 14, 2005.

[5] Mazur, Eric. "Farewell, lecture?" *Science* 323:5910, 2 January 2009, pp. 50–51.

[6] Berque, D. 2005. Promoting classroom interactivity in computer science courses using laptops, pen-based computers, Tablet PC's, and Dyknow software. J. Comput. Small Coll. 21:2 (Dec. 2005), 45-48.

[7] Griswold, W. G. and Simon, B. 2006. Ubiquitous presenter: fast, scalable active learning for the whole classroom. In Proceedings of the 11th Annual SIGCSE Conference on innovation and Technology in Computer Science Education (Bologna, Italy, June 26 - 28, 2006). ITICSE '06. ACM, New York, NY, 358-358. DOI= http://doi.acm.org/10.1145/1140124.1140268

[8] Pargas, R. P. 2006. Reducing lecture and increasing student activity in large computer science courses. In Proceedings of the 11th Annual SIGCSE Conference on innovation and Technology in Computer Science Education (Bologna, Italy, June 26 - 28, 2006). ITICSE '06. ACM, New York, NY, 3-7. DOI= http://doi.acm.org/10.1145/1140124.1140129.

[9] Roschelle, J.; Tatar, D.; Chaudbury, S.R.; Dimitriadis, Y.; Patton, C.; DiGiano, C. "Ink, improvisation, and interactive engagement: learning with tablets." *IEEE Computer* 40:9 (Sept. 2007), pp. 42–48.

[10] Angelo, Thomas A. and Cross, K. Patricia. *Classroom Assessment Techniques*, 2ed., Jossey-Bass, 1993.

Copyright © 2010, American Society for Engineering Education