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.

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.

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

| Google Docs - Folder - CSC | 517 - Mozilla Firefox | | |
|--|---|---|-------------------------------------|
| <u>File E</u> dit <u>V</u> iew Hi <u>s</u> tory <u>B</u> ook | marks <u>T</u> ools <u>H</u> elp | | 4 |
| 🔇 🔄 - C 🗙 🏠 | (http://docs.google.com/#folders/folder.0.0BxacXtA5Q8QcNTE4YjMxY | TYENWM1Ny00C 🏠 🔹 💽 🛛 G | oogle 🎾 |
| <u> Gmail</u> <u>Calendar</u> Documents | Reader Web more • ed.gehringer@gmail.com New! S | <mark>Share a folder</mark> <u>Offline</u> <u>Setti</u> | ngs <u>Help</u> <u>Sign out</u> |
| Google docs | Search Docs Search Templates | Show search options Browse template gallery | |
| Create new 👻 Upload | My folders > CSC 517 Share this folder - | | |
| Document | 🗹 🔻 Share 🕶 Move to 🔻 Folders 💌 Delete Rena | me More actions 🔻 1-50 of | 144 Last modified 🔻 |
| 📷 Presentation | S ² Name | Folders / Sharing | Date ↓ |
| 🌆 Spreadsheet | YESTERDAY | | |
| 🧓 Form | 🔲 🚖 🛛 👦 CSC 517 Exam Survey | John to 3 collaborators | Jan 5 me |
| 🧰 Folder | OLDER | and a discussion dist | 12/22/09 me |
| From template | Errors in automatically graded questions on | me to 1 collaborator, 1 viv me to 1 collaborator, 1 viv | 12/22/09 me |
| More searches - | Grading status | me to 2 collaborators | 12/14/09 John |
| wore searches + | Project-demo times signup | me to everyone, 1 collabo | 12/9/09 me |
| My folders | Classify your pattern as an object or class p | me to everyone, 1 collabo | 12/4/09 me |
| CSC 216 CSC 517 | Classify your pattern into one of these categories | me to everyone, 1 collabo | 12/4/09 me |
| Folders shared with me | Another design pattern | me to everyone, 1 collabo | 12/4/09 me |
| | 🔲 🏫 🔛 Project demo schedule | me to 6 collaborators | 12/4/09 Debadeep. |
| | Another example of Flyweight | me to everyone, 1 collabo | 12/4/09 me |

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.

| | - Google Docs - Mozilla Firefox |
|--------------------------------------|---|
| Eile Edit View History Bookm | arks Iools Help 🔅 |
| Google docs | ed.gehringer@gmail.com Help Sign out |
| 🕇 Add item 🔻 | Theme: Plain Email this form See responses • More actions • Save |
| Untitled fo | m |
| You can includ | e any text or info that will help people fill this out. |
| Question Title Help Text | Sample Question 1 |
| Question Type Their answer | Text |
| Sample Ques | |
| You can view the Ariewform?formka | published form here: http://spreadsheets.google.com y=dEY0ZjhDN05GRzA2OUZ2YUgzdjkyWFE6MA |

Figure 2. Editable boilerplate for a new Form

| Edit form - [Untitled Form] - Goog Ele Edit View History Bookmarks To | | Firefox | | | | |
|--|---|------------------------------|----------------------|------------------------|-----------------|--|
| < >>- C 🗙 🏠 🛅 htt | tp://spreadsheets.goo | gle.com/gform?key=tF4f8C7NFG | 069FvaH3v92XQ&hl=en; | # 🏠 • 💽 • Google | P | |
| Google docs | | | | ed.gehringer@gmail.com | Help Sign out | |
| Questions Text Paragraph text Multiple choice Checkboxes Choose from a list Scale Orid Other Section header Page break | t or info that will help Sample Question | | See responses + | More actions V Save |] | |
| Their answer | | | | | | |
| You can view the publishe Ariewform?formkey=dEY02 | | | | | | |

Figure 3. Adding a question to a form

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

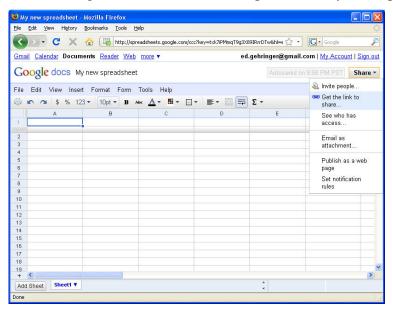


Figure 4. Sharing a document

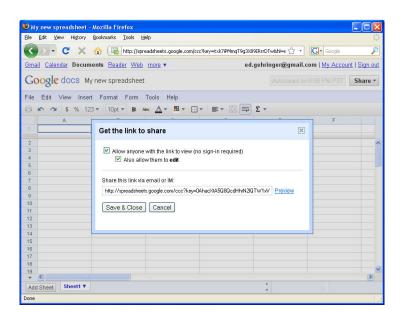
to use for an exercise: you need to make it editable by any student in your class (Figure 4). You do that by going to the Share menu and selecting, "Get the link to share". You will be required to save the document first. Then you need to check the two boxes to allow others to view the document without logging in, and to edit the document (Figure 5).

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.



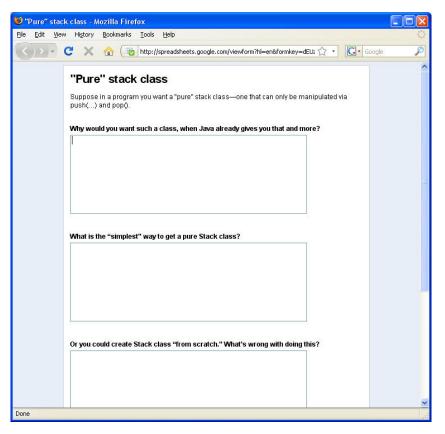
3. Uses of Google Forms

Figure 5. Making a spreadsheet editable by any student

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

Proceedings of the 2010 American Society for Engineering Education Annual Conference & Exposition Copyright © 2010, American Society for Engineering Education



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.

Figure 6. Collecting answers to prose questions

| Eile | Edit View History | / <u>B</u> ookmarks <u>T</u> ools <u>H</u> elp | | | | 3 |
|---------|---------------------------|--|---|---|--|-------------|
| < | D-CX | 🕜 🚺 http://spreadsheets.google.com | /ccc?key=0AhacXtA5Q8QcdEUzbVc1a3NNWDFT(| Q3RaWk1LTHUŁOWc&hl=ŧ ☆ ・ 💽 • Go | iogle | F |
| Gma | ail <u>Calendar</u> Docur | nents <u>Reader Web</u> more • | | ed.gehringer@gmail. | c om <u>My Account</u> <u>Sig</u> | <u>in o</u> |
| G | oogle docs "P | ure" stack class | | Autosave | d on 10/23/09 Sha | re |
| ile | Edit View Inse | rt Format Form (5) Tools Help | | | | |
| Ð | 🔊 🔉 💲 % 12 | 3 • 10pt • B. Ale A • 🗄 • 🕀 • | ≣• 🖾 🛱 Σ• | | | |
| | A | Ð | С | 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 | | 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 | | 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 | | 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 | | | |
| 7 | | | | | - | |
| 9 | | | | | | |
| 10 + | < | | | | | > |
| | d Sheet Sheet1 V | | | 8 | | - |

Figure 7. Spreadsheet populated as a form is submitted

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.

| Bite Edit Yew Higtory Bookmarks Tools Help | Bimplementation of addLast() in LinkedList class - Mozilla Firefox | |
|--|---|-------------------------------|
| Implementation of addLast() in LinkedList class Fill in the blanks in the implementation of addLast(). public void addLast(Object element) { | Eile Edit View History Bookmarks Iools Help | 0 |
| Implementation of addLast() in LinkedList class Fill in the blanks in the implementation of addLast(). public void addLast(Object element) { // Create the node to be added Node newNode = new Node(element, null); // It is empty, we just add new node 1st. // I See find end of list & add node there. Node current = first; while // See for addLast() Code for addLast() Your row 1 2 3 4 5 6 7 Front of classroom Implementation Implementation Implementation Implementation Submit Implementation Implementation Implementation Implementation | 💽 💽 - C 🗙 🏠 🕞 http://spreadsheets.google.com/viewform?hl=en&fo | rmkey=dGJmdXV: 🏠 🔹 🚺 Google 🖉 |
| Vone | Implementation of addLast() in Linke Fill in the blanks in the implementation of addLast() public void addLast(Object element) { // Create the node to be added Node newNode = new Node(element, null); // I' list is empty, we just add new node 1st. // GEmpty() { // I' list is empty, we just add new node 1st. // GEmpty() { // I' list is empty, we just add new node 1st. // GEmpty() { // I' list is empty, we just add node there. Node current = first; while (| dList class |

Figure 8. Form for having the students write code

| ≥ In ≅le | | ntation of View Histo | addLast() in LinkedList class - Mozilla Firefox v Bookmarks Tools Help | | | |
|-------------|----------|--------------------------|--|------------------------------|------------------|------------|
| 2 | | C > | | 5080cdG3md10/2/ 🔿 🔹 | G. Google | |
| | ul. Cala | | nents Reader Web more • | ed.gehringer@gm | | unt I Qian |
| _ | | | | | | |
| 60 | oogie | docs ir | plementation of addLast() in LinkedList class | Auto | saved on 11/5/09 | Share |
| ile | Edit | View Ins | rt Format Form (11) Tools Help | No other user | s viewing. | |
| 6 | 5 3 | | | Σ - | | |
| | Tim | Anestamp | B Code for addLast() | | C Your row | D |
| | | lestamp | Code for addLast() | | Tour row | |
| | | | public upid addl.act/Object.alament) (| | | |
| | | | public void addLast(Object element) { // Create the node to be added | | | |
| | | | Node newNode = new Node(element, null); | | | |
| | | | // If list is empty, we just add new node 1st. | | | |
| 2 | | | If (IsEmpty()) {first = newNode} // Else find end of list & add node there. | | | |
| | | | Node current = first; | | | |
| | | | while (this.next != null) current = next | | | |
| | | 11/3/2009 | current.next = newNode; | | | |
| | | 10:58:50 | | | 1 | |
| | | | public void addLast(Object element) { // Create the node to be added | | | |
| | | | Node newNode = new Node(element, null); | | | |
| | | | <pre>// If list is empty, we just add new node 1st. If (isEmpty()) {first=newNode; return}</pre> | | | |
| 3 | | | // Else find end of list & add node there. | | | |
| | | | Node current = first; | | | |
| | | | while (current.next != null) current=current.next: | | | |
| | | 11/5/2009 | current.next=newNode; | | | |
| | | 13:34:14 | | | 2 | |
| | | | public void addLast(Object element) { // Create the node to be added | | | |
| | | | Node newNode = new Node/element_null): | | | |
| + | < | 10 | | 3 | | 1 |
| Add | Sheet | Sheet1 V | | Timestar | np | |

Figure 9. Spreadsheet in which code appears

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.)

| 😕 Answers to JHavéPop exercises - Mozilla Firefox | | Inswers to JHavéPo | 2. | | | |
|---|------|--------------------------------------|----------------------------------|-----------------------|--|-------------------------|
| Eile Edit View Higtory Bookmarks Tools Help | Elle | Edit View History | <u>Bookmarks T</u> ool | s <u>H</u> elp | | |
| 🕢 🕞 🗶 🏡 🔯 http://spreadsheets.google.com/vik 🏠 🔹 💽 - Google 🔎 | < | D-CX | 🏠 🚺 http | ://spreadsheets.googk | e.com/cc 🏠 🔹 🚺 | • Google 🔎 |
| | Gm | ail <u>Calendar</u> Docum | nents <u>Reader We</u> | | | |
| Answers to JHavéPop exercises | | | | ed.gehringe | er@gmail.com <u>My Ac</u> | <u>count Sign out</u> |
| | G | oogle docs An | iswers to JHavéP | op exercise Aut | osaved on 6:24 PM PS | Share - |
| * Required | File | Edit View Inser | rt Format Form | (0) Tools Help | | |
| Your name/your partner's name * | | | | | · ≡• 🛛 🗖 | Σ - |
| | | A | B | С | D | E |
| | 1 | Timestamp | Your name/your partner's name | Question number | Your answer | |
| Question number * | 2 | 10/13/2009 11:00:59 | Student 1 | 1 | p = p.next(); | ^ |
| | з | 10/13/2009 11:01:12 | Student 1 | 2 | p = q; | |
| Your answer * | 4 | 10/13/2009 11:01:53 | Student 1 | 6 | p.info = r.info; | |
| | 5 | 10/13/2009 11:02:17 | Student 1 | F | p.info = p.next.next.info; | |
| | 6 | 10/15/2009 | Student 1 | | p.next = p.next.next; | |
| | 7 | 10/15/2009 10:04:12 10/15/2009 | Student 2 | 20 | q = p.next; p.next = null; r = q.next; q.next = p; p = r.next; r.next = q; p.next = r; | |
| Submit | 8 | 10:18:02 | Student 1 Student 3 / Student | 13 | while(!q.info.equals q = q.next; | |
| - Comme | 9 | 10:19:06 | | 3 | q=p; | |
| Powered by Google Docs | 10 | 10/15/2009 10:19:33 | Student 5 | 10 | p.next = q; | ~ |
| Report Abuse - Terms of Service - Additional Terms | 1 | id Sheet Sheet 1 V | |) St | udent 1 | |
| | | Find: | 📕 Next 🔿 | Previous 🖉 Highlig | | |
| Done | Done | | | | | |

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.

| . Dev. velovy | Bookmarks | Tools | Help | | | | |
|--|------------|-----------|----------------------|-----------------|------------------|---------------|-------------|
| - C × | ☆ (電 | http:// | (spreads | heets.go | ogle.com/viewfor | m?hl=er 🏠 🔹 | G• Google |
| Analysis o | of runn | ina t | ime | | | | |
| for selection so | | | | | | | |
| Use "^" for expon | | | | | | | |
| Cose Intexpoli | entration. | | | | | | |
| How many times | did we loo | k for the | e small | est eler | nent in the rest | of the array? | |
| | | | | | | | |
| la la china da di | | | | | | | |
| In looking for the | smallest e | element | , how n | nany itei | ms did we need | to examine, | on average? |
| | | | | | | | |
| | | | | | | | |
| So, how many tir | nes did we | have to | look a | t an elei | ment? | | |
| So, how many tir | nes did we | have to |) look a | t an elei | ment? | | |
| So, how many tir | nes did we | have to |) look a | t an elei | ment? | | |
| So, how many tir | | | | | | | |
| | | | | | | | |
| Therefore, the ar | | | | | | | |
| | nount of w | ork rise | s prop | ortional | to | | |
| Therefore, the ar | nount of w | ork rise | s prop 4 5 | ortional 6 7 | to | room | |
| Therefore, the a | nount of w | ork rise | s prop 4 5 | ortional 6 7 | to | room | |
| Therefore, the ar Your row Front of classroo | nount of w | ork rise | s prop 4 5 | ortional 6 7 | to | room | |
| Therefore, the ar | nount of w | ork rise | s prop 4 5 | ortional 6 7 | to | room | |

Figure 11. Asking for the steps in a derivation

| mail Calendar Documents Reader Web more ▼ ed.gehringer@gmail.com Mr Account Sign c Coogle docs Analysis of running time Autosaved on 11/24/09 Share le Edit View Insert Format Form (12) Tools Help E F le Edit View Insert Format Form (12) Tools Help E F le Edit View Insert Format Form (12) Tools Help E F le Edit View Insert Format Form (12) Tools Help E F le We look for the smallest element, how many times did we look for the smallest element, how many times did we look for the smallest element, not an element? So, how many times Therefore, the did we have to look at amount of work rises an element? Your row 11/2/2/2009 n-1 n^2-2n+1 1 n^2-2n+1 6 11/2/2/2009 n-1 n^2-2n+1 1 n^2-2n+1 6 11/2/2/2009 n-1 (n+1)/2 element is the number of elements. 6 11/2/2009 n-1 (n-1)! (n-1)(n-1)! ((n-1)*2)! 1 | | Edit View History | | | | | | 0 |
|--|-----|-------------------------|--|--|-----------------------------|-------------------------|-----------------------------------|----------|
| A B A B C D E F A B C D E F | 5 | C X | 👔 🏠 🕼 http://spreadsh | neets.google.com/ccc?key=0Ahac | XtA5Q8QcdEdyR2lRZXhFbms2Uzd | 2ay1tczh6N1E: 🎧 🔹 | .oogle | ۶ |
| Edit View Insert Format Form (12) Tools Help A B C D E F A B C D E F A B C D E F How many times did we look for the smallest element in the rest of the array? In looking for the smallest element, on average? So, how many times did we have to look at an element? Therefore, the amount of work rises Your row 11/24/2009 13:30:01 n-1 n*2-2n+1 1 n*2-2n+1 6 11/24/2009 13:30:51 n-1 (n+1)/2 element is the number of elements. 6 11/24/2009 13:30:51 n-1 (n-1)1 (n-1)(n-1)1 ((n-1)^h2)1 1 11/24/2009 15:01:34 6 1 1 1 1 11/24/2009 15:01:34 6 1 1 1 1 11/24/2009 15:01:34 6 1 1 2 1 11/24/2009 15:02:24 n n n*2-n n*2 6 11/24/2009 15:02:35 n n n 1 2 1 11/24/2009 15:03:16 | mai | <u> Calendar</u> Docum | nents <u>Reader Web</u> more | | | ed.gehringer@gmail.c | om <u>My Account</u> <u>S</u> | lign ou |
| Edit View Insert Format Form (12) Tools Help A B C D E F A B C D E F A B C D E F How many times did we look for the smallest element in the rest of the array? In looking for the smallest element, on average? So, how many times did we have to look at an element? Therefore, the amount of work rises Your row 11/24/2009 13:30:01 n-1 n*2-2n+1 1 n*2-2n+1 6 11/24/2009 13:30:51 n-1 (n+1)/2 element is the number of elements. 6 11/24/2009 13:30:51 n-1 (n-1)1 (n-1)(n-1)1 ((n-1)^h2)1 1 11/24/2009 15:01:34 6 1 1 1 1 11/24/2009 15:01:34 6 1 1 1 1 11/24/2009 15:01:34 6 1 1 2 1 11/24/2009 15:02:24 n n n*2-n n*2 6 11/24/2009 15:02:35 n n n 1 2 1 11/24/2009 15:03:16 | -0 | ogle docs Ar | alysis of running time | | | | on 11/24/09 Sh | nare 🔻 |
| A B C D E F A B C D E F How many times did we look for the smallest element in the rest of the array? In looking for the smallest element in the rest of the array? So, how many times did we have to look at an element? Therefore, the did we have to look at an element? Your row 11/24/2009 n-1 n*2-2n+1 1 n*2-2n+1 6 11/24/2009 n-1 (n+1)/2 element is the number of elements. 6 11/24/2009 n-1 (n-1) (n-1)(n-1)((n-1)((n-1)/2)) 1 1 11/24/2009 n-1 n^2 1 1 1 13/30:53 n-1 (n-1)! (n-1)(n-1)! (n-1)(n-1)! 1 11/24/2009 5 5 3 3 2 1 11/24/2009 n n n 1 2 1 11/24/2009 5 3 333 20 n*2 6 11/24/2009 5 3.333 20 | | U | | | | | | |
| A B C D E F How many times did we look for the smallest element in the rest of the array? In looking for the smallest element in the rest of the array? So, how many times did we have to look at an element? Therefore, the did we have to look at an element? Your row 11/24/2009 13:30:51 11/24/2009 13:30:53 13:30:53 13:30:53 11/24/2009 15:02:17 11/24/2009 15:02:17 11/24/2009 15:02:24 11/24/2009 15:02:24 11/24/2009 15:02:24 11/24/2009 15:02:24 11/24/2009 15:02:24 11/24/2009 15:02:24 11/24/2009 15:02:24 11/24/2009 15:02:26 11/24/2009 15:02:26 11/24/2009 15:02:26 11/24/2009 15:02:26 11/24/2009 15:02:26 11/24/2009 15:02:36 12:02:36 12:02: | ile | Edit View Inse | rt Format Form (12) To | ols Help | | | | |
| How many times did we look for the smallest element, how many items did we need to examine, on average? So, how many times an element? Therefore, the amount of work rises proportional to Your row 11/24/2009 13:30:51 11/24/2009 13:30:51 11/24/2009 13:30:55 11/24/2009 13:30:55 11/24/2009 15:01:34 11/24/2009 15:01:34 11/24/2009 15:01:34 11/24/2009 15:02:59 11/124/2009 15:02:59 12 12 12 12 12 12 12 12 12 12 12 12 12 | | \$ % 1 2 | 23 • 10pt • B . A⊯ <u>A</u> • | | Ξ Σ - | | | |
| How many times did we look for the smallest element in the rest of the array? smallest element, we meed to examine, on average? So, how many times did we have to look at amount of work rises an element? Your row 11/24/2009 13:30:02 n-1 n^2-2n+1 1 n^2-2n+1 6 11/24/2009 13:30:51 n-1 n^2-2n+1 1 n^2-2n+1 6 11/24/2009 13:30:53 n-1 (n+1)/2 element is the number of elements. 6 11/24/2009 13:30:53 n-1 (n-1)! (n-1)(n-1)! ((n-1)^2)! 1 11/24/2009 15:01:34 5 n n n 1 2 1 11/24/2009 15:02:17 N-1 N*(N-1)/2 - - 2 1 11/24/2009 15:02:24 N-1 N*(N-1)/2 - - 2 1 11/24/2009 15:02:26 n-1 n n - - 2 1 11/24/2009 15:03:16 N=1 N*(N-1)/2 - - 2 1 11/24/2009 15:03:16 N=1 N*(N-1)/2 N*(N-1)/2 N*(N-1)/2 <t< th=""><th></th><th>A</th><th>В</th><th>С</th><th>D</th><th>E</th><th>F</th><th></th></t<> | | A | В | С | D | E | F | |
| 11/24/2009 n-1 n^2-2n+1 1 n^2-2n+1 6 11/24/2009 N+N-1+N-2+,+1 (n+1)/2 element is the number of elements. 6 11/24/2009 n-1 (n-1)! (n-1)(n-1)! ((n-1)^n2)! 1 13/30:53 n-1 (n-1)! (n-1)(n-1)! ((n-1)^n2)! 1 11/24/2009 n-1 n n-1 n^2 1 11/24/2009 1 5 | 1 | Timestamp | we look for the smallest element in | smallest element, how many items did we need to examine, | did we have to look at | amount of work rises | Your row | |
| 133002 h-1 h*2-2h+1 1 h*2-2h+1 6 11/24/2009 1330:51 N+N-1+N-2++1 (n+1)/2 depends on where the element is 6 11/24/2009 1330:53 n-1 (n-1)! (n-1)(n-1)! ((n-1)*2)! 1 11/24/2009 n n n-1 n*2 1 11/24/2009 5 11/24/2009 1 1 15/02:17 N-1 N*(N-1)/2 2 1 11/24/2009 5 1 2 1 15/02:17 N-1 N*(N-1)/2 2 2 11/24/2009 1 n/2 n*2 6 11/24/2009 1 n/2 - 2 11/24/2009 1 1 1 2 11/24/2009 1 1 1 2 11/24/2009 1 1 1 1 2 11/24/2009 5 3.333 20 n*2 5 5 11/24/2009 5 3.333 20 n*2 5 5 11/24/2009 1 1 1 1 < | 2 | | | | | | | |
| 13:30:51 N+N-1+N-2++1 (n+1)/2 element is the number of elements. 6 11/2/2009 n-1 (n-1)! (n-1)(n-1)! ((n-1)^2)! 1 13:30:53 n-1 (n-1)! (n-1)(n-1)! ((n-1)^2)! 1 13:31:57 n n n-1 n*2 1 11:24/2009 1 11:24/2009 1 1 15:01:34 5 3 2 1 11:24/2009 n n/2 - 2 11:24/2009 n n n/2 - 6 11:24/2009 n n n - 1 2 11:24/2009 1 5 3.333 20 n*2 6 1 11:24/2009 1 5 3.333 20 n*2 6 1 | | | n-1 | n^2-2n+1 | | n^2-2n+1 | 6 | |
| 11/24/2009 11/24/2009 13/30/53 n-1 11/24/2009 n 13/31/57 n 11/24/2009 n 15/01/34 5 11/24/2009 1 15/01/34 5 11/24/2009 1 15/02:17 N-1 N*(N-1)/2 n^2 15/02:24 n-1 n/2 n^n2 11/24/2009 n 15/02:24 n-1 n/2 -n^2 11/24/2009 1 15/02:24 n-1 n/2 -n^2 11/24/2009 1 15/02:24 n-1 n n*2 11/24/2009 1 15/02:24 n-1 n n*2-n 11/24/2009 5 15/03:16 N-1 N*(N-1)/2 N*(N-1)/2 11/24/2009 1 15/03:16 N-1 N*(N-1)/2 N*(N-1)/2 1 1 1 1 | 3 | | NUN 1 (N O) 11 | (0.11)() | | the number of elements | 0 | |
| 13:30:53 n-1 (n-1)! (n-1)(n-1)! ((n-1)*2)! 1 11:724/2009 n n n-1 n*2 1 11:724/2009 5 11:724/2009 2 1 15:01:34 5 2 1 2 11:724/2009 1 1 1 2 11:724/2009 1 1 2 1 11:724/2009 1 1 2 1 11:724/2009 1 1 2 1 11:724/2009 1 1 1 2 11:724/2009 1 1 1 1 11:724/2009 1 1 1 1 11:724/2009 1 1 1 1 11:724/2009 5 3:333 20 1 11:724/2009 5 3:333 20 1 11:724/2009 1 1 1 1 11:724/2009 1 1 1 1 11:724/2009 1 1 1 1 1 | - | | NTN-ITN-ZTTI | (11+1)/2 | CICILICIILIS | the number of elements. | 0 | - |
| 11/24/2009 n n n-1 n ⁿ 2 1 13/31:57 n n n n-1 n ⁿ 2 1 13/31:57 n n n n-1 n ⁿ 2 1 15/01:34 5 - - 2 11/24/2009 15/02:17 N-1 N*(N-1)/2 2 2 11/24/2009 15/02:24 n-1 n/2 - n ⁿ 2 6 11/24/2009 15/02:59 n-1 n n ⁿ 2-n n ⁿ 2 6 11/24/2009 15/03:01 5 3.333 20 n ⁿ 2 5 11/24/2009 1 5 3.333 20 n ⁿ 2 5 11/24/2009 5 3.333 20 n ⁿ 2 5 5 11/24/2009 5 3.333 20 n ⁿ 2 5 5 11/24/2009 1 N*(N-1)/2 N*(N-1)/2 2 2 11/24/2009 1 1 N*(N-1)/2 1 2 11/24/2009 1 1 N*(N-1)/2 1 2 11/24/2009 1 1 1 | 1 | | n-1 | (n-1)! | (n-1)(n-1)! | ((n-1)^2)! | 1 | |
| 1333157 n n n n-1 n*2 1 111/24/2009 5 | 5 | | | | | | | |
| 15:01:34 5 11/24/2009 N*(N-1)/2 15:02:17 N-1 15:02:17 N-1 15:02:17 N-1 15:02:24 n-1 15:02:59 n-1 11:724/2009 1 15:03:16 N-1 N*(N-1)/2 (N-1)/2 11:724/2009 1 11:724/2009 1 15:03:16 N-1 N*(N-1)/2 N*(N-1)/2 11:724/2009 1 11:724/2009 1 11:724/2009 1 11:724/2009 1 11:724/2009 1 11:72 | | | n | n | n-1 | n^2 | 1 | |
| 11/24/2009 N*(N-1)/2 2 11/24/2009 n-1 n/2 ~n^2 n ⁴ 2 6 11/24/2009 15:02:24 n-1 n/2 ~n ⁴ 2 6 11/24/2009 15:02:59 n-1 n n ⁴ 2-n n ⁴ 2 11/24/2009 15:03:01 5 3.333 20 n ⁴ 2 6 11/24/2009 15:03:01 5 3.333 20 n ⁴ 2 6 11/24/2009 15:03:16 N-1 N*(N-1)/2 (N-1)/2 N*(N-1)/2 2 | | | 5 | | | | | |
| 15:02:17 N-1 N*(N-1)/2 2 11:02:17 N-1 n/2 ~n^2 n 15:02:24 n-1 n/2 ~n^2 n^2 6 11:02:29 n n n^2-n n^2 6 11:02:29 n n n^2-n n^2 6 11:02:29 n n n^2-n n^2 6 11:02:209 1 n n*2-n n*2 5 11:02:209 1 5 3.333 20 n*2 5 11:02:209 1 1:03:16 N-1 N*(N-1)/2 (N-1)/2 N*(N-1)/2 2 1:03:16 N-1 N*(N-1)/2 (N-1)/2 N*(N-1)/2 2 2 | | | J | | | | | - |
| 15:02:24 n-1 n/2 ~n^2 n^2 6 11:02:29 n-1 n n^2-n n^2 6 15:02:59 n-1 n n^2-n n^2 6 11:02:59 n-1 n n^2-n n^2 6 11:02:59 n-1 n n^2-n n^2 6 11:02:10 5 3.333 20 n^2 6 11:03:16 N-1 N*(N-1)/2 (N-1)/2 N*(N-1)/2 2 | | | N-1 | N*(N-1)/2 | | | 2 | |
| 16:02:24 n-1 n/2 ~n*2 n*2 6 11/24/2009 n n n*2-n n*2 6 15:02:59 n-1 n n n*2-n n*2 6 11/24/2009 5 3.333 20 n*2 5 11/24/2009 5 3.333 20 n*2 5 11/24/2009 15:03:16 N-1 N*(N-1)/2 (N-1)/2 N*(N-1)/2 2 | | | | | | | | |
| 15:02:59 n-1 n n²-n n²2 11:1/24/2009 1 5 3.333 20 n²2 5 11:1/24/2009 1 5 3.333 20 n²2 5 11:1/24/2009 15:03:16 N*(N-1)/2 (N-1)/2 N*(N-1)/2 2 | | | n-1 | n/2 | ~n^2 | n^2 | 6 | <u> </u> |
| 11/24/2009 5 3.333 20 n*2 6 11/24/2009 11/24/2009 1 5 5 11/24/2009 1 1 N*(N-1)/2 N*(N-1)/2 2 | | | n_1 | n | n^9_n | nA9 | | |
| 15:03:01 5 3.333 20 n^2 6 11:/24/2009 N*(N-1)/2 (N-1)/2 N*(N-1)/2 2 11:/24/2009 N*(N-1)/2 (N-1)/2 N*(N-1)/2 2 | | | 11-1 | 11 | 11 2-11 | 11 2 | | - |
| 15:03:16 N-1 N*(N-1)/2 (N-1)/2 2 11/24/2009 | E | | 5 | 3.333 | 20 | n^2 | 5 | |
| 15:03:16 №1 №*(№-1)/2 (№-1)/2 №*(№-1)/2 2 11/24/2009 | 1 | | | | | | | |
| | - | | N-1 | N*(N-1)/2 | (N-1)/2 | N*(N-1)/2 | 2 | |
| | | | | | | | | > |
| | | | | | | Timestamn | | |

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.

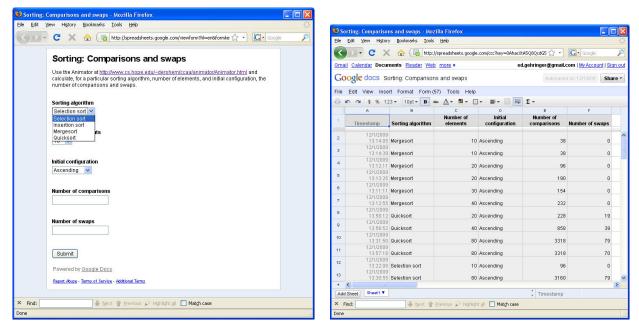


Figure 13. Working cooperatively with a sort simulator to produce a table of sorting times

| - C X 🏠 (🔯 http://spreadsheets.google.com/viewform?hl=ent 🏠 🔹 💽 🕻 Google | | lassify your patte | rn as an object o | class pattern - I | Aozilla Firefox | | |
|---|------|-------------------------|----------------------------------|---------------------|------------------------|--|-------------------|
| Classify your pattern as an object or class pattern | Ele | <u>E</u> dit ⊻jew Higto | ry <u>B</u> ookmarks <u>T</u> oo | ils <u>H</u> elp | | | |
| | | D. C) | < 🏠 🔚 http | ://spreadsheets.goo | gle.com/ccc?key=0Ahac) | (tA5080cdGlc 1/2 + | G . Google |
| * Required | Gma | il Colondar Deci | ments Reader W | | | ed.gehringer@gmail. | |
| Pattern name * | | | | | | a.geni inger @gmail. | .com (my Account |
| Abstract Factory | GC | ogle docs o | lassify your patte | rn as an object o | or class pattern | Autosav | ved on 12/4/09 |
| Category | File | Edit View Ins | ert Format Form | (8) Tools Help | | | |
| Creational V | 6 | n n \$ % 1 | 23 - 10pt - B. | Анк <u>А</u> - 🖽 - | | Σ- | |
| Annana and a second and a second | | A | в | c | D | E | F |
| Object or class? | 1 | Timestamp | Pattern name | Category | Object or class? | Comments | Your location |
| Object 🛩 | = 2 | 12/4/200 | | | | Refers to an object and provides an | ť. |
| | 2 | | i Iterator | Structural | Object | order structure. | Monteith 313 |
| Comments | | | | | | encapsulates all data of an object | |
| | 3 | 12/4/200 | | | | for later method | |
| | | 13:58:4 | Command | Behavioral | Object | call. | Monteith 313 |
| | 4 | 14:02:3 | Abstract Factory | Creational | Object | | Monteith 313 |
| | 5 | 12/4/200 14:03:2 | a 2 Interpreter | Behavioral | Class | | Monteith 313 |
| | 0 | 12/4/200 | | Oburtural | Class | | Monteith 313 |
| | 7 | 12/4/200 | a i | Structural | Class | | Monteith 313 |
| | | 14:04:3 | 2 Decorator | Structural | Object | | |
| Our location | 8 | 14:05:0 | 2 Façade | Structural | Class | | |
| O EB 1220 | 9 | 12/4/200 |) 7 Flyweight | Behavioral | Object | | Monteith 313 |

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.

| | | eresting facts - Mozilla Firefox kmarks <u>T</u> ools <u>H</u> elp | |
|---|---|---|--|
| Coole does Java UI Component — Interesting fact Management — Interesting fact (or method) 1 Management — Interesting fact (or method) 1 10/1/2009 Checkbox Checkbox All or any combination of boxes can be A picture can be added beside the list of boxes To make a window that is dependent on another window — disappearing when the other window, use an internal frame. Capable of making frames within frames, called internal frames. Capable of making frames within frames, called internal frames. Also able to make frames that allow scrolling and tabbed frames. The default cell renderer knows how to display strings and icons and it displays Objects by invoking to String | 🔇 🔊 - C 🗙 🏠 | http://spreadsheets.google.com/ccc?key=0AhacX | ASQ8QcdEp10XBoSGdCLU1LeFM2UUNCc1QxWr ☆ - 💽 · Google 🖉 |
| File Site Vew Instance Format | <u>imail Calendar</u> Documents <u>Reader</u> ⊻ | Neb more V | ed.gehringer@gmail.com <u>My_Account</u> <u>Sign out</u> |
| Image: Section of the sectin of the section of the | Google docs Java UI Compone | nts — interesting facts | Autosaved on 10/1/09 Share - |
| Image stamp Uf component the you chose Interesting fact (or method) 1 Interesting fact (or method) 2 10/1/2009 All or any combination of boxes can be 13:45:35 All or any combination of boxes can be checked A picture can be added beside the list of boxes 10/1/2009 The dimensions of the border area may 10:42:40 Frame A picture can be added beside the list of boxes 10/1/2009 The dimensions of the border area may 10:42:40 Frame Capable of making frames within frames, called internal frames. This allows for one frame. Also able to make frames that allow scrolling and tabbed frames. 10/1/2009 can select multiple others within the frame. Also able to make frames that allow scrolling and tabbed frames. 10/1/2009 can select multiple items or combination frame. The default cell renderer knows how to display strings and icons and it displays Objects by invoking to String 10/1/2009 Menu as a popup menu. As you can see, a menu item can have either an image or text, or both. 10/1/2009 as a popup menu. It has methods: getUl() getUlClassID() | ïle Edit Mew Insert Format Form | n (27) Tools Help | |
| Uit component that you there the provided the state of the provided the state of the provided the state of the provided the provided the state of the provided the provided the provided the state of the provided the pro | 🕽 🖛 🕿 💲 🐒 123 - 12pt - 🖪 | <u>⊷ ∆</u> - ∎- ⊡- ≡- ⊡ ≡ Σ- | |
| 13:45:35 Checkbox checked A picture can be added beside the list of boxes 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. Capable of making frames within frames, called internal frames. This allows for one frame, then multiple others within the frame. 10/1/2009 13:45:41 Frame Capable of making frames within the frame, then multiple others within the frame. 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.) 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 13:44:36 JMenu Menu As you can see, a menu item can have either an image or text, or both. It has methods: getUl() getUlClassID() | UI component that you | t | |
| 10/1/2009 10:42:40 The dimensions of the border area may be obtained using the getInsets method. Capable of making frames within frames, called internal frames. This allows for one frame, then multiple others within the frame. 10/1/2009 13:45:41 Frame 10/1/2009 13:44:28 JList Of items (i.e. item 0, item 2, item 16 etc.) 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. Imestry | | | A picture can be added beside the list of boxes |
| 10/1/2009 13:45:41 10/1/2009 13:45:41 Frame 10/1/2009 13:44:28 JList Called internal frames. This allows for one frame, then multiple others within the frame. 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.) 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: getUl() getUlClassID() | 10/1/2009 | | disappearing when the other window is iconified, for example — use a dialog instead of frame To make a window that appears |
| 10/1/2009 13:44:28 JList can select multiple items or combination of items (i.e. item 0, item 2, item 16 etc.) and it displays Objects by invoking to String 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. 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() | | called internal frames. This allows for one frame, then multiple others within the frame. | Also able to make frames that allow scrolling and tabbed |
| 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. 10/1/2009 13:44:36 JMenu JMenu As you can see, a menu item can have either an image or text, or both. It has methods: getUl() getUl(ClassID() | | can select multiple items or combination | |
| It has methods: getUl() getUlClassID() | | they aren't placed with the other components in the UI. Instead, a menu usually appears either in a menu bar or | |
| Add Steet Timestamp | | ας α μομαρ πιθημ. | It has methods: getUl() |
| (meaning) | and the second se | | * Timestamp |
| | | 🞩 Next 🔶 Previous 🖉 Highlight all 🔲 Match case | e miestany |
| Done | 1976) | Town Freedom Freedom Freedom | |

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.

| 🕲 Rubric areas for program style - Mozilla Firefox | 🕑 Advice on a particular rubric area - Mozilla Firefox |
|--|---|
| Elle Edit View History Bookmarks Iools Help | Elle Edit View History Bookmarks Iools Help |
| 🔇 💽 - C X 🏠 (👼 http://spreadsheets.google.com/viewform?hl=en&form 🏠 - 🔀 - Google 🔎 | 🔇 💽 - C X 🏠 🔞 http://spreadsheets.google.com/viewform?hl=en&for 😭 - 🖸 🗘 - Google 🔎 |
| Rubric areas for program style Here are some areas in which we might grade your programs for style. Each one is followed by several questions. Please make gygestions about what should be included in the questions, or ways in which the questions might be worded. | Advice on a particular rubric area Please provide guidance on how to grade a program on one of the topics covered in the rubric. * Required |
| Formatting Is code properly and consistently indented? Are braces used and located consistently? Is whitespace used for readability? Are too-long-lines generally avoided? | Area * What area (formatting, naming, commenting, visibility, structure, and testing) are you providing advice for? Formatting |
| | 5. Exemplary What are the characteristics of work that merits a "5" in this category? |
| Naming Are names meaningful, and suggestive of the purpose for which the name is used? | 4. Proficient What are the characteristics of work that merits a "4" in this category? |
| Commenting Are comments used to explain what each class does? Do all nonaccessor methods have comments gwing the purpose, name, and parameter of the method? Are comments used to explain non-obtaous sertions of code within methods? Done | × Find: ▶ Next |

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.

| W CRC cards for flight-reservation system - Mozilla Firefox | | | | | | | | |
|--|-------------|-----------------------------------|-------------------|-----------|---------------------|----------------|---------|--------------------------|
| Elle Edit View Higtory Bookmarks Iools Help | | | | | | | | |
| 🕜 💽 🕫 🗶 🏠 🛅 http://spreadsheets.google.com/ccc?key=0AhacXtA5Q8QcdC1TRXIxdUdQWTFOb2RGWjBiWURHU 🏠 🔹 💽 🕞 Google | | | | | | | | |
| Gmail Calendar Documents Reader Web more • ed.gehringer@gmail.com My Account Sign out | | | | | | | | |
| Google docs CRC cards for flight-reservation system Autosaved on 10/7/09 Share - | | | | | | | | |
| | | | | | | | | |
| File Edit View Insert Format Form Tools Help | | | | | | | | |
| 💿 🗠 🔉 % 123 τ 10pt τ 🖪. 🛶 🛆 τ 🖽 τ 🖽 τ 🚍 τ Ξ 🚍 Σ τ | | | | | | | | |
| | A | B | С | D | E | F | G | н |
| 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 | | | | | |
| 22 | customers | issue itinerary to | anport | | | | | |
| 5 | airport | passengers | flight | | | | | |
| 6 | | | | Customer | Source, Destination | | | |
| 7 | customer | know the route | agent | Customer | List of Passengers | Passenger | | |
| 8 | | provide flight and | | | | | | |
| 9 | ticket | itinerary details | | airplane | offer seating map | customer | | |
| 10 | lionor | | | anpiano | cher county map | ou ou on on ou | | |
| 11 | | | | customer | pick a seat | airplane | | |
| 12 | | | | | | plane ticket | | |
| 13 | | - | | - | Know the origin | | | |
| 14 | | | | Itinerary | Origin | Passenger | Flight | Know the plane number |
| | Reservation | - | | lunerary | Know the | Passenger | riigiit | Know the flight |
| 15 | System | search for flights | search engine | | destination | Passenger | | number |
| 16 | | 10 000 m | The second second | | | | | Know the flight |
| | System | generate ticket | ticket_generator | | | | | length |
| 17 18 | Sytem | email | emailer | | | | | |
| 19 | | | | | | | | |
| 20 | | | | Passender | Know his/her name | | | ~ |
| | | | | | | | | > |
| Add Sheet V Class | | | | | | | | |
| 🗙 Find: 🖉 Vext 👚 Previous 🖉 Highlight all 🗌 Match case | | | | | | | | |
| Done | | | | | | | | |
| | | | | | | | | |

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.

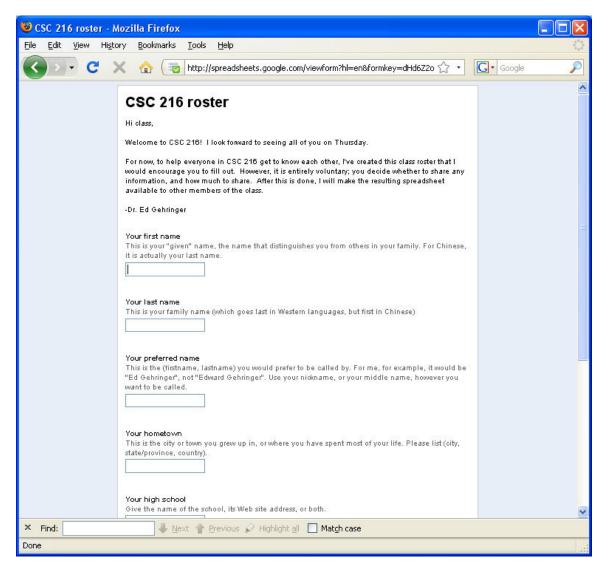


Figure 18. Class roster

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.

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

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 <u>http://tinyurl.com/gdoc-use-describe</u>. Responses of others will be viewable at <u>http://tinyurl.com/gdoc-use-view</u>.

Bibliography

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

[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= <u>http://doi.acm.org/10.1145/1140124.1140129</u>.

[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.