Peer Instruction Intro

2001-11-19

⇐ **Dan Garcia**
(www.cs.berkeley.edu/~ddgarcia)

Dave Patterson ⇒
(www.cs.berkeley.edu/~patterson)

3 Ideas on Improving Learning

1. Peer Instruction

- In class testing of concepts via multiple choice that answer individually 1st then after discussing in groups vote as a team

2. Just in time teaching

- Students read material, answer questions on web form (including what don't understand) BEFORE lecture so instructor can emphasize what students are having trouble with

3. Electronic voting: poll a class in seconds anonymously (or not) and plot results

° 61C tried all 3; you might mix and match

New Experiment: Peer Instruction

- **Increase real-time learning in lecture, test understanding of concepts vs. details**
  `mazur-www.harvard.edu/education/pi.phtml`

- **As complete a "segment" ask multiple choice question**

  - **1-2 minutes: decide yourself, vote**

  - **2-3 minutes: discuss in pairs, then team vote; flash cards/PRS**

    - Try to convince partner; learn by teaching

- **But how include concept tests yet still cover all material?**
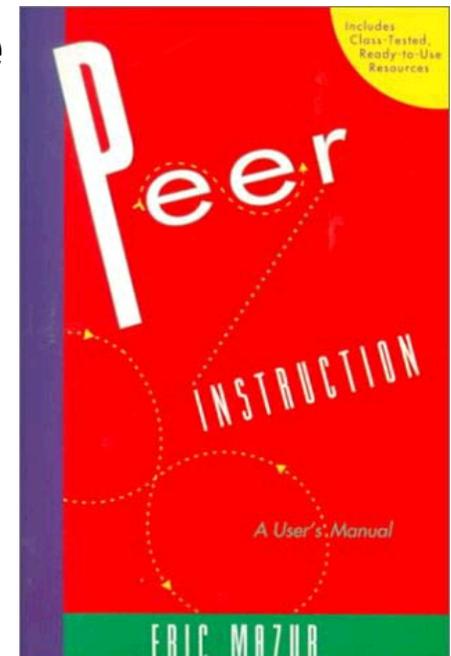
Just-In-Time-Teaching

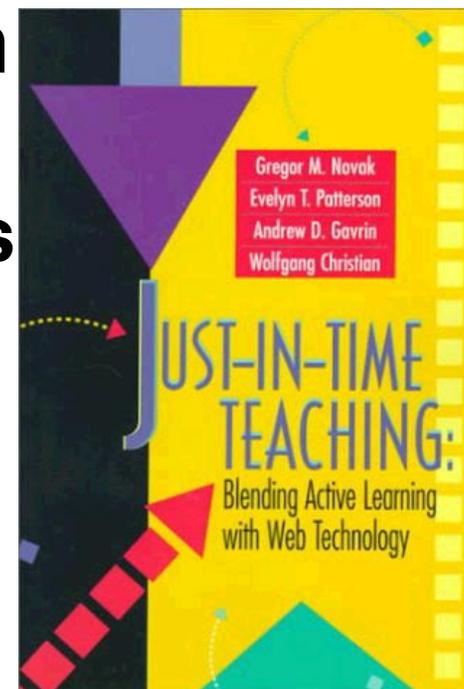○ **Must** read textbook and review lectures notes **before class** "Just-in-Time Teaching" `webphysics.iupui.edu/jitt/jitt.html`

- **Reduces examples have to do in class**

- **Get more from lecture (also good advice)**

○ **Fill out 3 question Web Form on (deadline 2AM before lecture)**

- **Graded for effort, not correctness**

- **5% of total grade**

"The students come to class prepared and already engaged with the material, and the faculty member already knows exactly where the students are and where classroom time together can be best spent."

Gregor M. Novak
Evelyn T. Patterson
Andrew D. Gavrin
Wolfgang Christian

JUST-IN-TIME TEACHING: Blending Active Learning with Web Technology

Technology for anonymous in-class voting

- PRS: Hardware + Software for student voting: collects votes and plots results in seconds
  `www.educue.com/Home.htm`

- Plug receivers into any laptop

- Initial plan is to equip several large lecture halls with receivers

- Students buy transmitters from bookstore (buy for ~ $30, sell back for ~ $15)

- Reuse in many classes (Chem 1A, Bio 1A, CS 61C, ...)

Quantitative Evaluation of Peer Instruction

"Peer Instruction: Ten Years of Experience and Results," Catherine H. Crouch and Eric Mazur, *Am. J. Phys.* 69, 970-977 (2001).

- standardized tests, repeated finals before and after PI show large gains

"Peer Instruction: Results from a Range of Classrooms,"Adam P. Fagen, Catherine H. Crouch, and Eric Mazur, submitted to *Phys. Teach.*

- 384 instructors try PI, 90% use again
- 11 courses using standardized tests before and after show large gains

Berkeley Midterm Survey results 61C

○ **Peer Instruction: In class concept test**
  - I really enjoy and learn from it: 53%
  - on the whole I prefer: 32%
  - on the whole I'd rather we skipped: 8%
  - I really dislike it: 7%

○ **Reading Quiz: Just in time learning**
  - I'm glad; I get more from lecture: 27%
  - on the whole a good idea: 51%
  - on the whole a bad idea: 13%
  - The quiz is unnecessary: 10%

○ **Do the lectures cover too much?**

  - Yes: 10%; No: 85%; Sometimes: 5%

Reading Quiz Example: Online by 2AM

° **First reading assignment for Friday:**
COD: 4.1, 4.2, 4.3, 4.4; K&R: 2.2, 2.3

1. Why do computers and the C language have both signed and unsigned integers? What would be wrong with just having signed integers in the computer? in C? (Your answer to each question should be 1 paragraph long.)

2. Given that Moore's Law doubles the number of transistors every 18 months, someone could build a decimal number computer. What would be the advantages and disadvantages of such a computer? What might a computer-user notice about a decimal computer? What might a programmer notice?

3. What did you find difficult or confusing about the reading? If nothing was difficult or confusing, tell us what you found most interesting. Please be as specific as possible.

Peer Instruction Example

```
int main(void){
  int A[2], *ptr;
  ptr=A;
  A[0]=5; A[1]=10;
  printf( %u %d %d %d\n ,ptr,*ptr,A[0],A[1]);
  ptr = ptr + 1;
  printf( %u %d %d %d\n ,ptr,*ptr,A[0],A[1]);
  *ptr = *ptr + 1;
  printf( %u %d %d %d\n ,ptr,*ptr,A[0],A[1]);
}
```
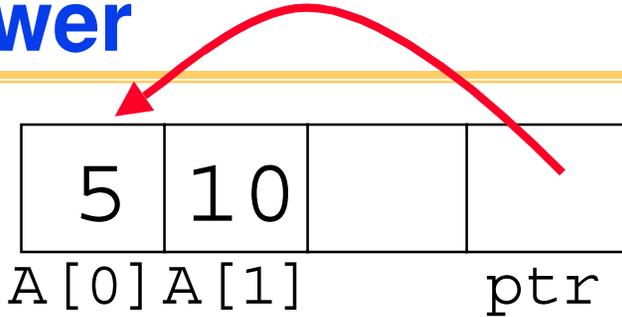
5	10		

A[0] A[1]   ptr
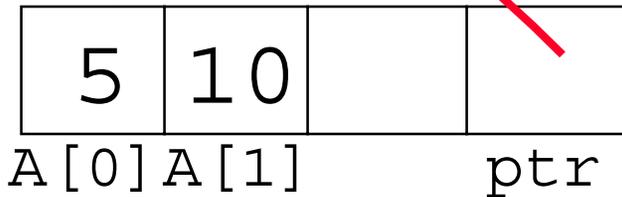
If the first `printf` outputs 100 5 5 10, **what will the other two `printf` output?**

```
1:  101 10 5 10        then 101 11 5 11
2:  104 10 5 10        then 104 11 5 11
3:  101 <other> 5 10   then 101 <3-others>
4:  104 <other> 5 10   then 104 <3-others>
5:  One of the two printfs causes an ERROR
6:  I surrender!
```

Peer Instruction Answer

```
int main(void){
  int A[2], *ptr;
  ptr=A;
  A[0]=5; A[1]=10;
  printf( %u %d %d %d\n ,ptr,*ptr,A[0],A[1]);
  ptr = ptr + 1;
  printf( %u %d %d %d\n ,ptr,*ptr,A[0],A[1]);
  *ptr = *ptr + 1;
  printf( %u %d %d %d\n ,ptr,*ptr,A[0],A[1]);
}
```

5	10		

A[0] A[1]   ptr

If the first `printf` outputs <u>100 5 5 10</u>, what will the other two `printf` output?

2: 104 10 5 10   then 104 11 5 11

…because `ints` in this system are 4-bytes long and the actual address increments by 4 even though it appears to only incrememt 1.