Peer Instruction for the First Time:

Experiences of a First Time User in Computer Science

Beth Simon
Computer Science and Engineering
University of California, San Diego

Formerly,
STLF Computer Science
CWSEI
UBC

UCSDCSE
Computer Science and Engineering

CWSEI
Carl Wieman Science Education Initiative
at the University of British Columbia

By the end of today's talk you should be able to…

- LG1: Describe to a colleague what "ConcepTests"/clicker questions took the form of in introductory computing
- LG2: Assess the impact of clicker questions in your class via analysis of normalized gain based on student solo and group vote
- LG3: Discover easy ways to find great exam questions at the last minute
- LG4: Quote yet another set of rave student reviews for use of peer instruction in the classroom
- LG5: Feel more comfortable being a "first timer" with peer instruction in your class

Tried to Implement Peer Instruction "Faithfully"

- Assigned Reading Before Class
- Posed Solo/Group Vote questions
  - X% of time
  - Judged time based on perception of student engagement
  - Solo: timed
  - Discussion: not timed (with iclicker clock)
- Always indicated correct answer and tried to always discuss wrong answers
- Tried to always get students to explain a rationale why someone would guess a wrong answer
- Frequently and repeatedly motivated to students why clickers help them learn
  - Need to make a list of what I actually said and did and how often

Tried to Implement Peer Instruction "Faithfully", BUT…

- No reading quizzes
  - And reading assignments not "spot on"
- Didn't "assign" discussion groups (tried for 4 people last term)
- No lecture at all (basically)
  - The book is not hard to read
  - Concepts don't rely on previous knowledge (e.g. calculus)
  - Book has good examples
- Did supplement with some meta-issues (not REALLY in book) (Note: I'd like to code these into categories - SIGCSE)
  - Debugging
  - Explanations of why
  - Memory model diagrams
  - Meta-analysis of how code works

1) Solo: (30 sec)
2) Discuss: (1min)
3) Group: (30 sec)

How many times is each set of code executed?

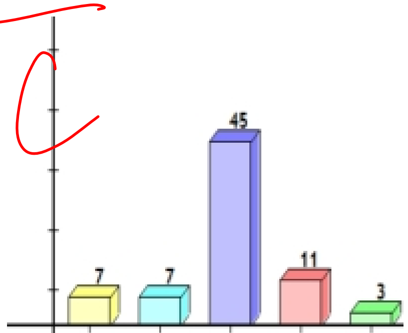
```
Pixel[] pixelArray = this.getPixels();
int value = 0;
Pixel p  = null;
for(int index = 0; index < pixelArray.length; index++);
{
  p = pixelArray[index];
  value = p.getRed();
  value = (int) (value * 0.5);
  p.setRed(value);
}
```

A

B

C

D

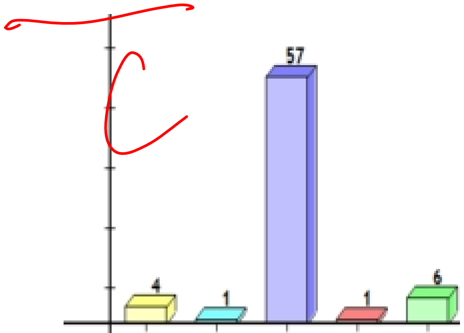


	Sect A	Set B	Set C	Set D
A	1	1	1	Many
B	1	Many	1	many
C	1	many	many	Many
D	Many	Many	Many	Many

1) Solo: (30 sec)
2) Discuss: (1min)
3) Group: (30 sec)

How many times is each set of code executed?

```
Pixel[] pixelArray = this.getPixels();
int value = 0;
Pixel p  = null;
for(int index = 0; index < pixelArray.length; index++);
{        A
  p = pixelArray[index];          B          C
  value = p.getRed();
D value = (int) (value * 0.5);
  p.setRed(value);
}
```

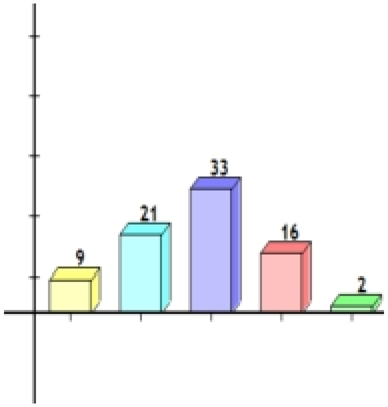


	Set A	Set B	Set C	Set D
A	1	1	1	Many
B	1	Many	1	many
C	1	many	many	Many
D	Many	Many	Many	Many

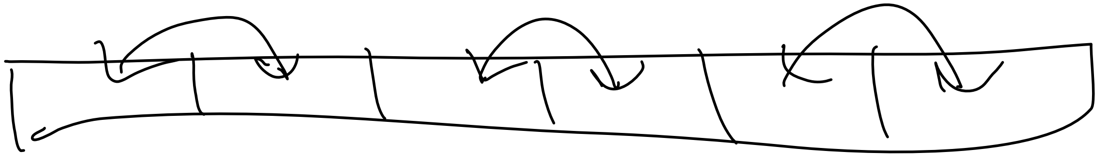
1) Solo: (30 sec)
2) Discuss: (2min)
3) Group: (30 sec)

What picture most accurately describes what this code does ?

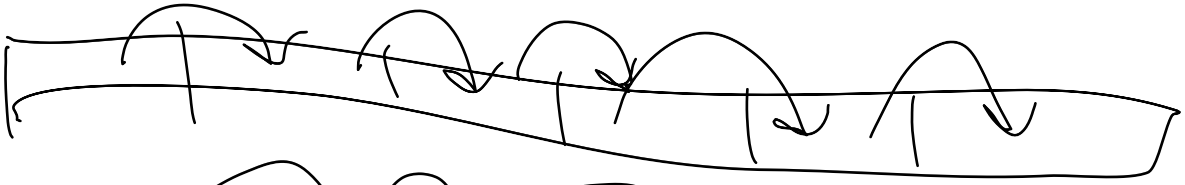
```
Pixel[] pixelArray = this.getPixels();
int value = 0;
Pixel p  = null;
for(int index = 0; index < pixelArray.length;
{
  p = pixelArray[index];
  q = pixelArray[index+1];
  p.setRed(q.getRed());
  p.setBlue(q.getRed());
  p.setGreen(q.getGreen());
}
```



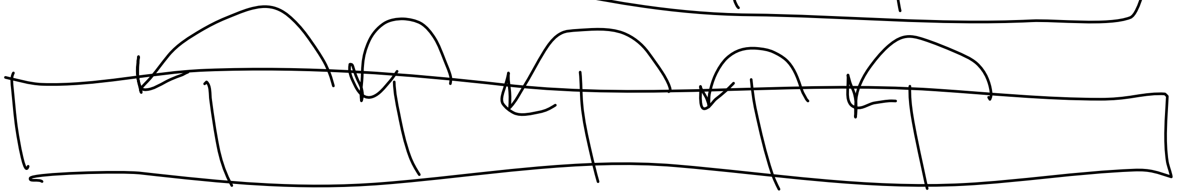
A)



B



C

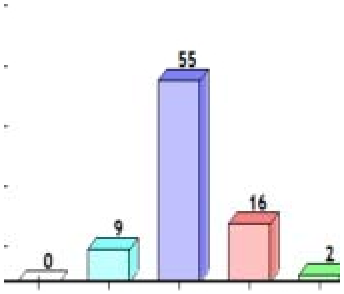


D. None of these.

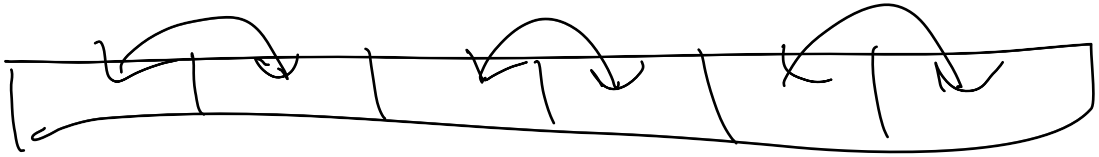
1) Solo: (30 sec)
2) Discuss: (2min)
3) Group: (30 sec)

What picture most accurately describes what this code does ?

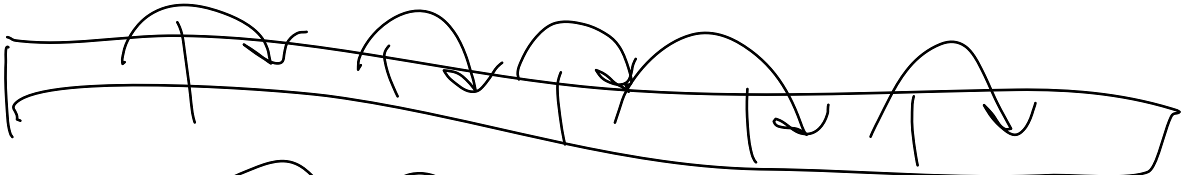
```
Pixel[] pixelArray = this.getPixels();
int value = 0;
Pixel p  = null;
for(int index = 0; index < pixelArray.length; index++)
{
  p = pixelArray[index];
  q = pixelArray[index+1];
  p.setRed(q.getRed());
  p.setBlue(q.getRed());
  p.setGreen(q.getGreen());
}
```



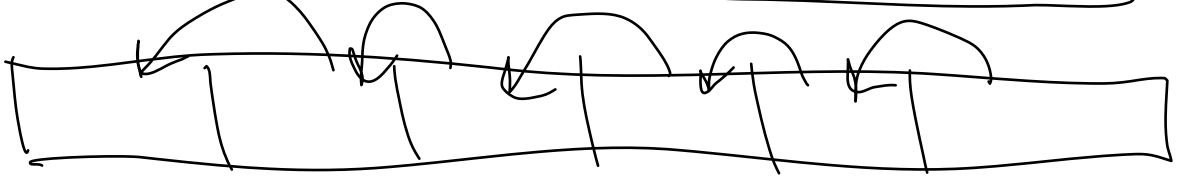
A)



B



C



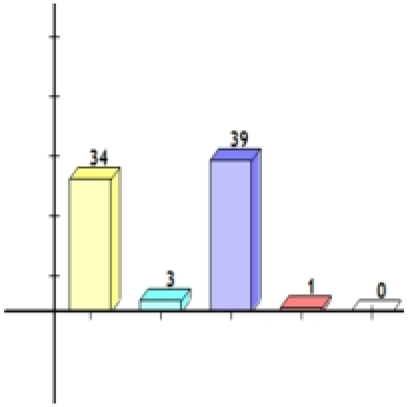
D. None of these.

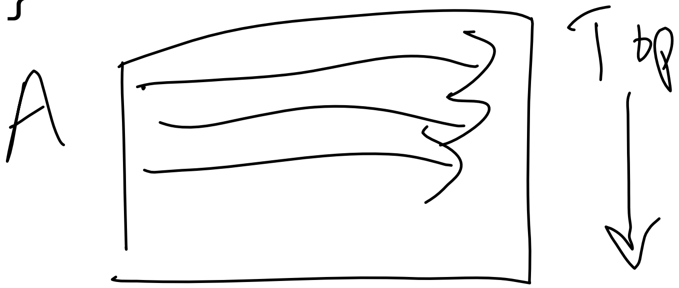
1) Solo: (30 sec)
2) Discuss: (2min)
3) Group: (30 sec)

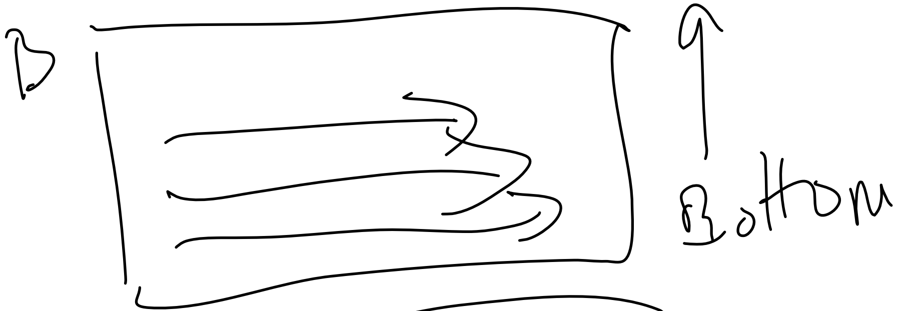
Nested Loops: How do they work? What order are pixels changed?

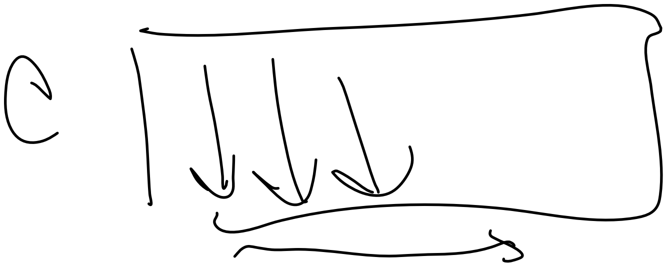
- A method in Picture.java…

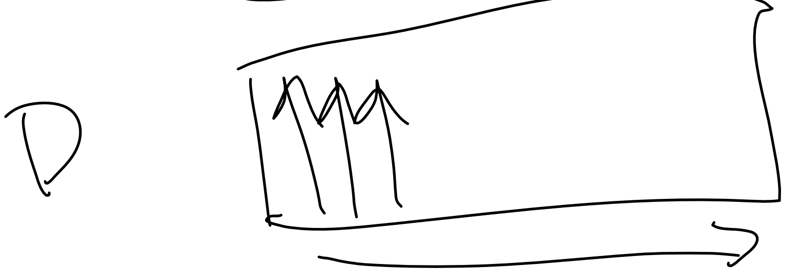
```
Pixel p;
for (int foo = 0; foo < getWidth();  foo++)
{
  for (int bar 0; bar < getHeight(); bar++)
  {
     p = getPixel(foo, bar);
     p.setColor(Color.BLACK);
  }
}
```









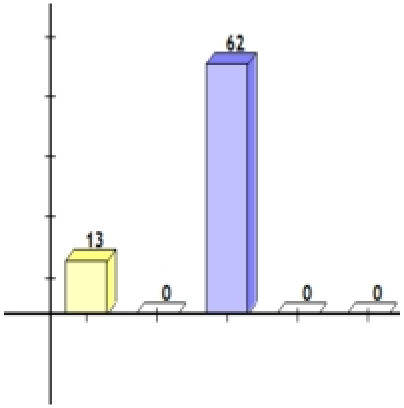


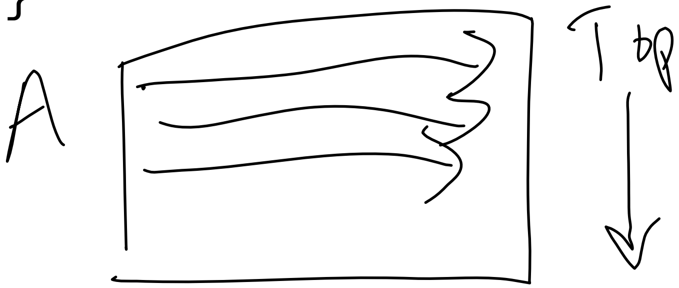
1) Solo: (30 sec)
2) Discuss: (2min)
3) Group: (30 sec)

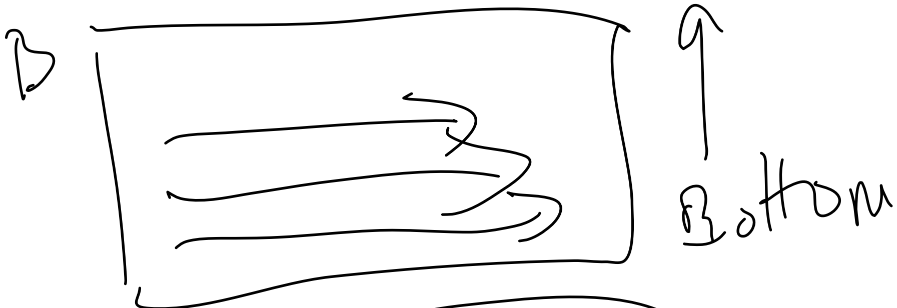
Nested Loops: How do they work? What order are pixels changed?

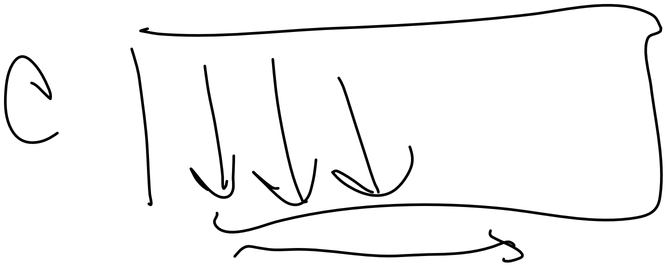
- A method in Picture.java…

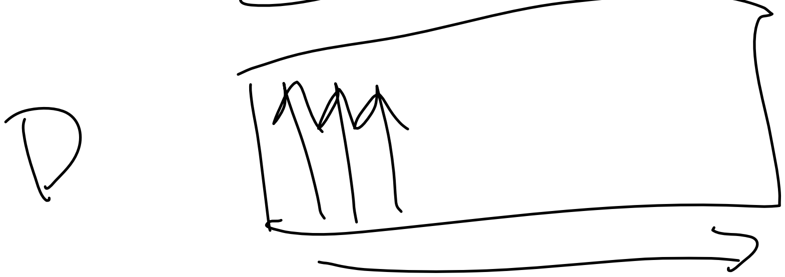
```
Pixel p;
for (int foo = 0; foo < getWidth();  foo++)
{
  for (int bar 0; bar < getHeight(); bar++)
  {
     p = getPixel(foo, bar);
     p.setColor(Color.BLACK);
  }
}
```







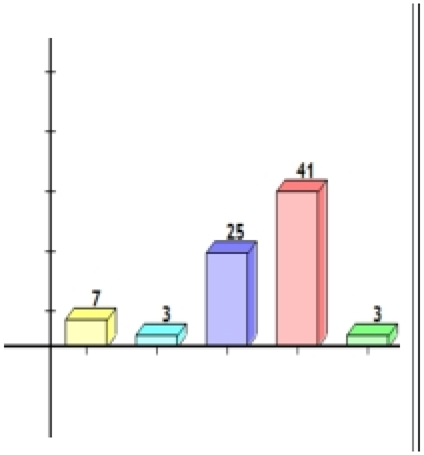




1) Solo: (30 sec)
2) Discuss: (2min)
3) Group: (30 sec)

Why does this code have an error?

```
Pixel[] pixelArray = this.getPixels();
int value = 0;
Pixel p  = null;
for(int index = 0; index < pixelArray.length; index++)
{
  p = pixelArray[index];
  q = pixelArray[index+1];
  p.setRed(q.getRed());
  p.setBlue(q.getRed());
  p.setGreen(q.getGreen());
}
```



A. It tries to access pixelArray[-1]

B. It tries to access pixelArray[0]

C. It tries to access pixelArray[pixelArray.length]

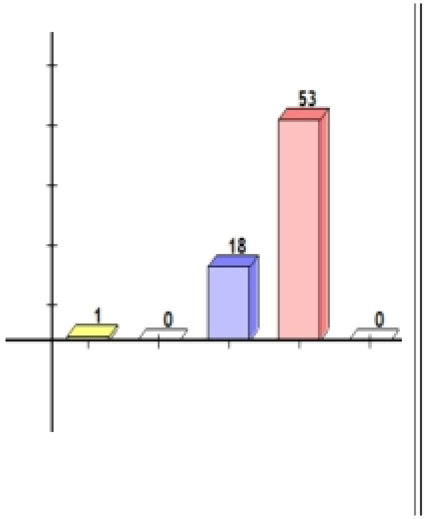
D. It tries to access pixelArray[pixelArray.length+1]

E. None of the above

1) Solo: (30 sec)
2) Discuss: (2min)
3) Group: (30 sec)

Why does this code have an error?

```
Pixel[] pixelArray = this.getPixels();
int value = 0;
Pixel p  = null;
for(int index = 0; index < pixelArray.length; index++)
{
  p = pixelArray[index];
  q = pixelArray[index+1];
  p.setRed(q.getRed());
  p.setBlue(q.getRed());
  p.setGreen(q.getGreen());
}
```



A. It tries to access pixelArray[-1]

B. It tries to access pixelArray[0]

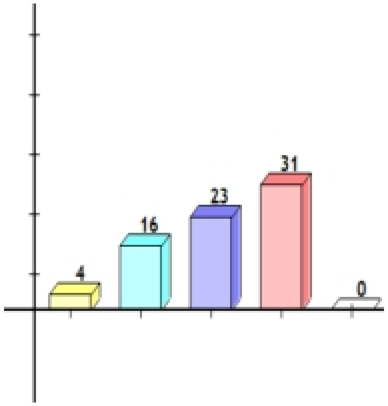
C. It tries to access pixelArray[pixelArray.length]

D. It tries to access pixelArray[pixelArray.length+1]

E. None of the above

1) Solo: (30 sec)
2) Discuss: (1min)
3) Group: (30 sec)

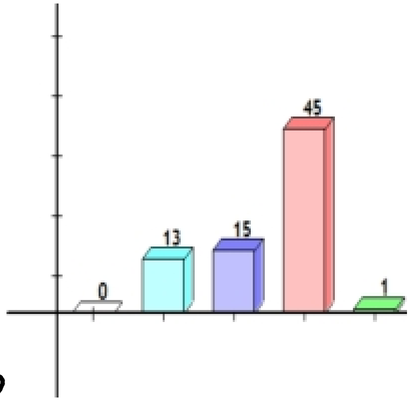
DEBUGGING: Where is the best place to put a print statement?



```
Pixel[] pixelArray = this.getPixels();          A
int value = 0;
int index = 0;
while (index < pixelArray.length)               B
{                                                 C
  value = pixelArray[index].getRed();
  value = (int) (value * 0.5);
  pixelArray[index].setRed(value);
  index = index + 1;                              D
}
```

1) Solo: (30 sec)
2) Discuss: (1min)
3) Group: (30 sec)

DEBUGGING: Where is the best place to put a print statement?

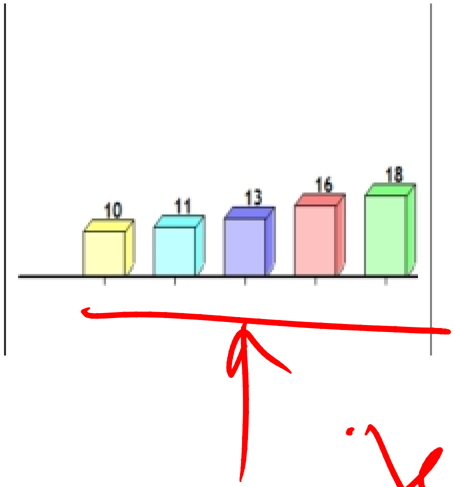


```
Pixel[] pixelArray = this.getPixels();
int value = 0;
int index = 0;                                          A
while (index < pixelArray.length)                       B
{                                                       C
  value = pixelArray[index].getRed();
  value = (int) (value * 0.5);
  pixelArray[index].setRed(value);
  index = index + 1;                                    D
}
```

1) Solo: (20 sec)
2) Discuss: (1 min)
3) Group: (20 sec)

SPEEDY: What does that code do?

```
public void funky()
{
  SoundSample[] noiseArray = this.getSamples();
  int zzz = 0;
  for (int i=0;i<noiseArray.length)
  {
    SoundSample sample = noiseArray[i];
    int foo = sample.getValue();
    zzz += foo;
  }
  int yyy = zzz / noiseArray.length;
  for (int i = 0; i < noiseArray.length; i++)
  {
    SoundSample sample = noiseArray[i];
    sample.setValue(yyy);
  }
}
```



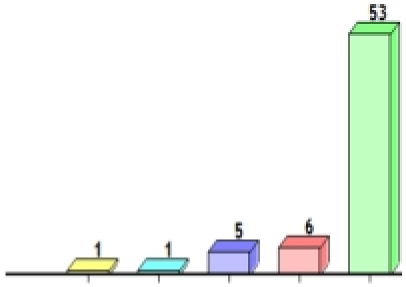
Favorite of Students spread!!

A. Makes it higher pitched
B. Makes it louder
C. Makes if lower pitched
D. Makes it quieter
E. Makes a silent sound

1) Solo: (20 sec)
2) Discuss: (1 min)
3) Group: (20 sec)

SPEEDY:
What does that code do?

```
public void funky()
{
  SoundSample[] noiseArray = this.getSamples();
  int zzz = 0;
  for (int i=0;i<noiseArray.length)
  {
    SoundSample sample = noiseArray[i];
    int foo = sample.getValue();
    zzz += foo;
  }
  int yyy = zzz / noiseArray.length;
  for (int i = 0; i < noiseArray.length; i++)
  {
    SoundSample sample = noiseArray[i];
    sample.setValue(yyy);
  }
}
```

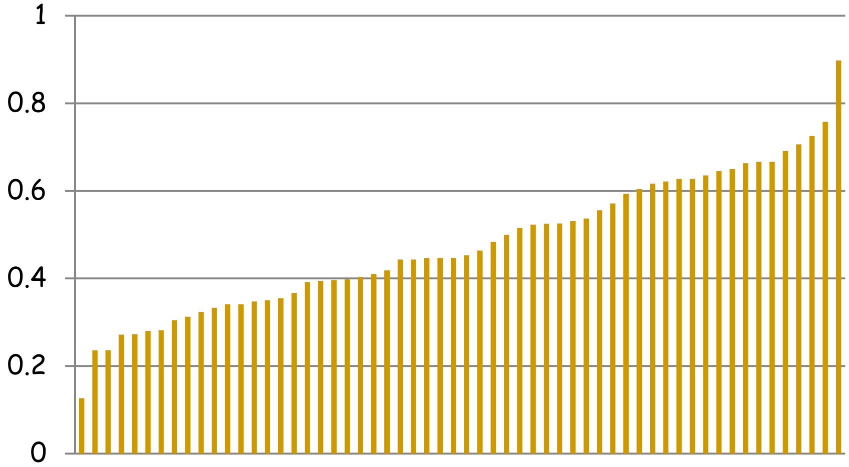


A. Makes it higher pitched
B. Makes it louder
C. Makes if lower pitched
D. Makes it quieter
E. Makes a silent sound

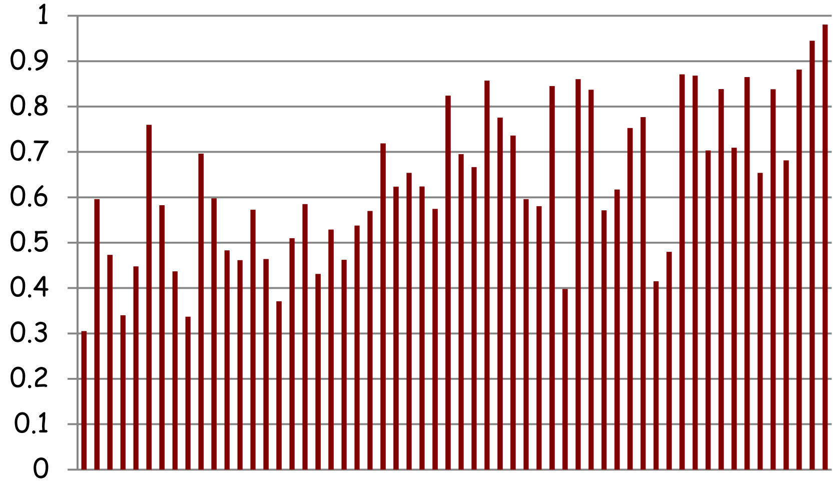
How did they do?

CS1 Fall 2008

Solo Correctness: avg 48%

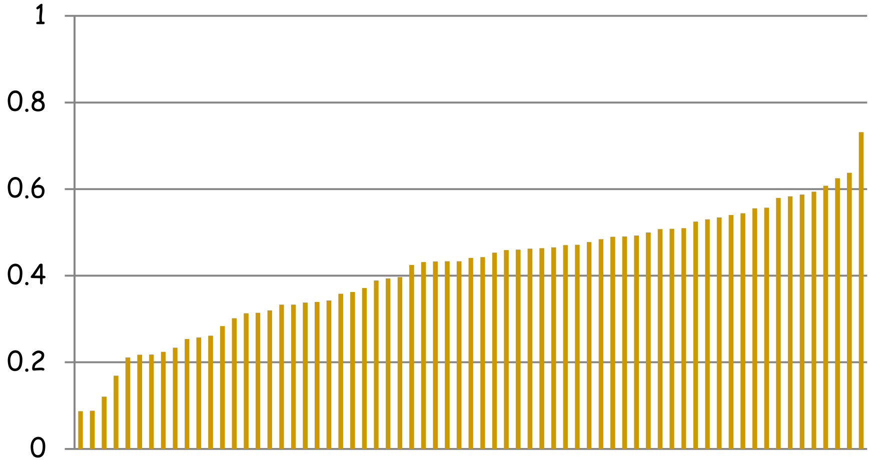


Group Correctness: avg 64%

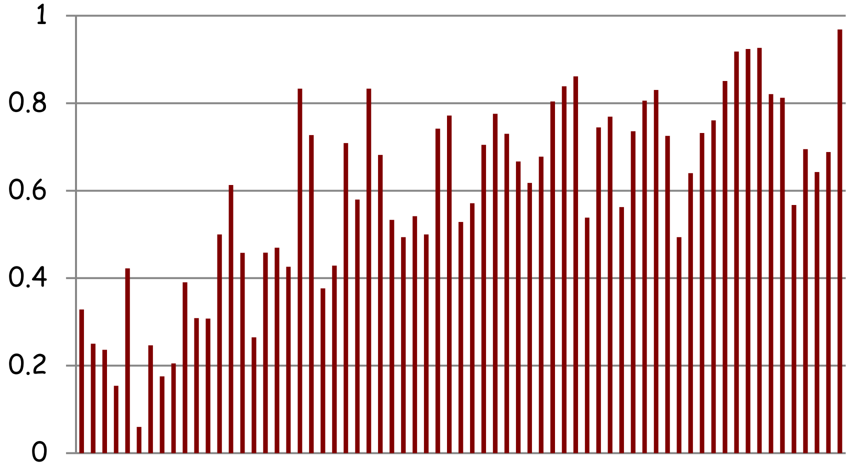


CS1.5 Winter 2009

Solo Correctness: avg 41%



Group Correctness: avg 60%

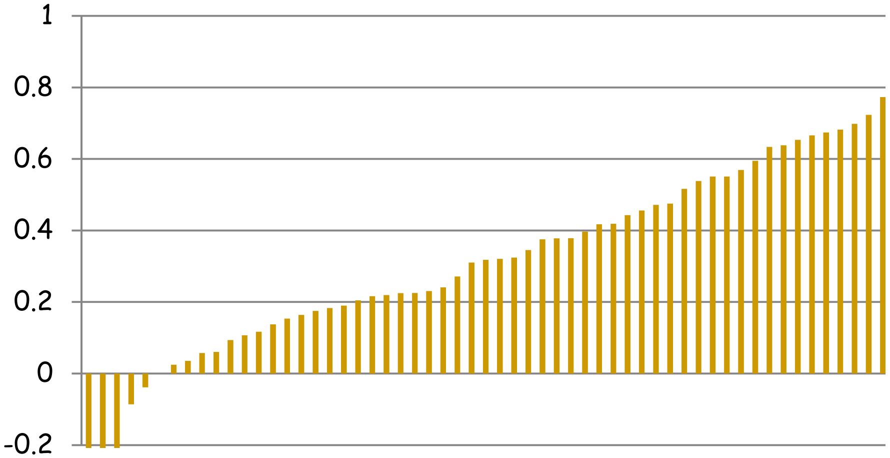


* Mazur: >35% && <70% solo

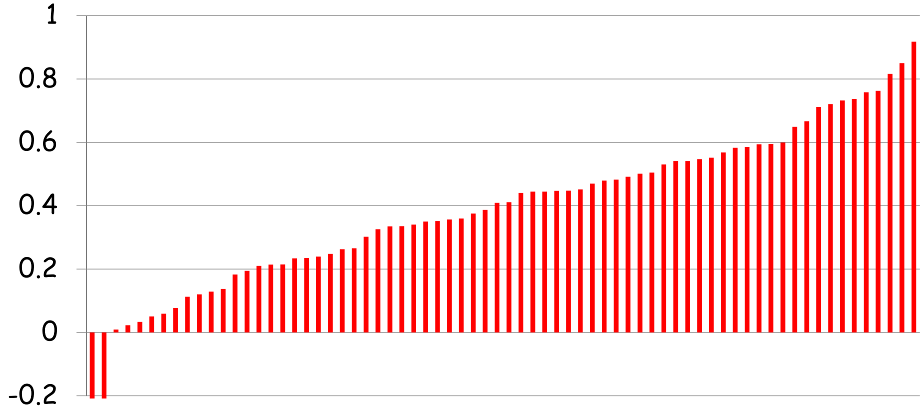
* Genetics: 52% solo, 68%group

Normalized Gain: (group-solo)/(1-solo)

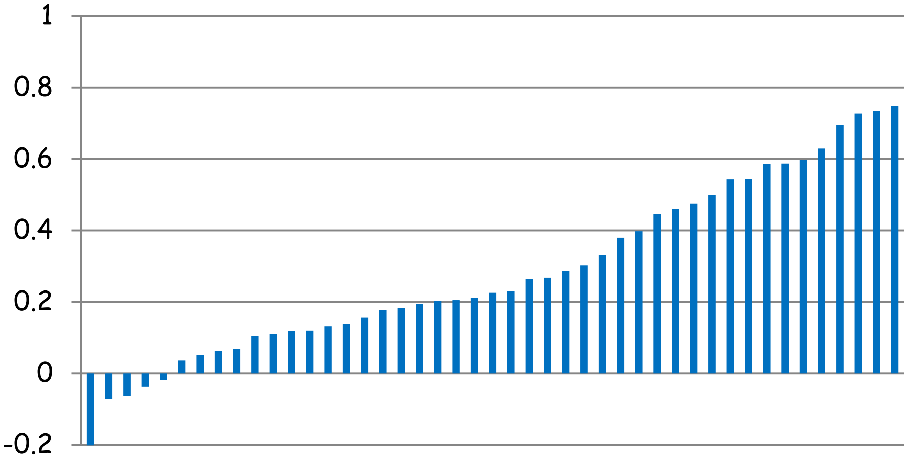
CS1 Fa 2008: avg 31%



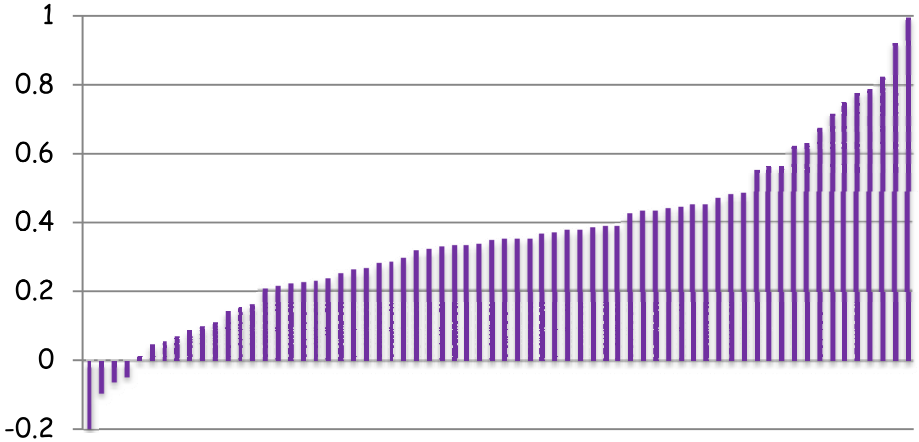
CS1 Wi 2009: avg 39%



CS1.5 Wi 2009: avg 29%



CS1.5 Sp 2009 - avg 35%



*Genetics course at UC: 41%-48% (unpublished)

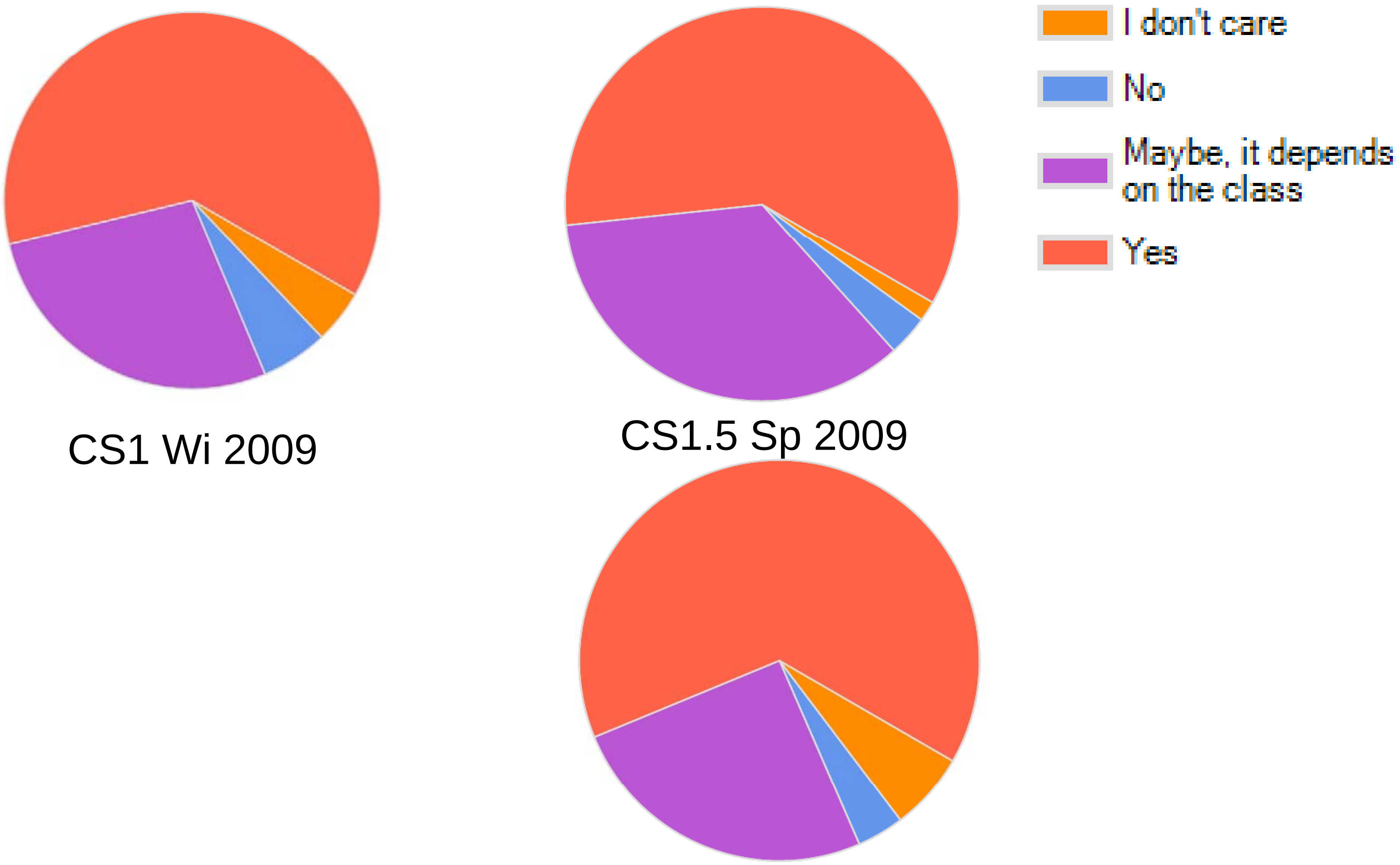
Making a midterm:

- A new course
  - CS1 in fall 2008, media computation
  - Whew! Review clicker questions, ask those
- CS 1.5
  - Specifically LOOK for clicker questions where the group vote was poor
    - Use identical or ISOMORPHIC questions

CS1.5 Spring 2009
Clicker Performance vs. Exam Performance

Content	Solo (avg =)	Group (avg =)	Exam
Integer parameter pass	34%	60%	97%, 95%
Object parameter pass	56%	49%	97%
Object new, param pass	53%	74%	83%
Array list remove and size (tricky) None of the above	24%	23%	84%
Ordering of exception catch	19%	26%	59%
Array element underpinning of ArrayList (mod clicker: add, exam: remove)	23%	60%	77% (84% if index off by one, but right pattern)

Would you recommend that other instructors in computing courses use clickers with discussion?



CS1 Wi 2009

CS1.5 Sp 2009

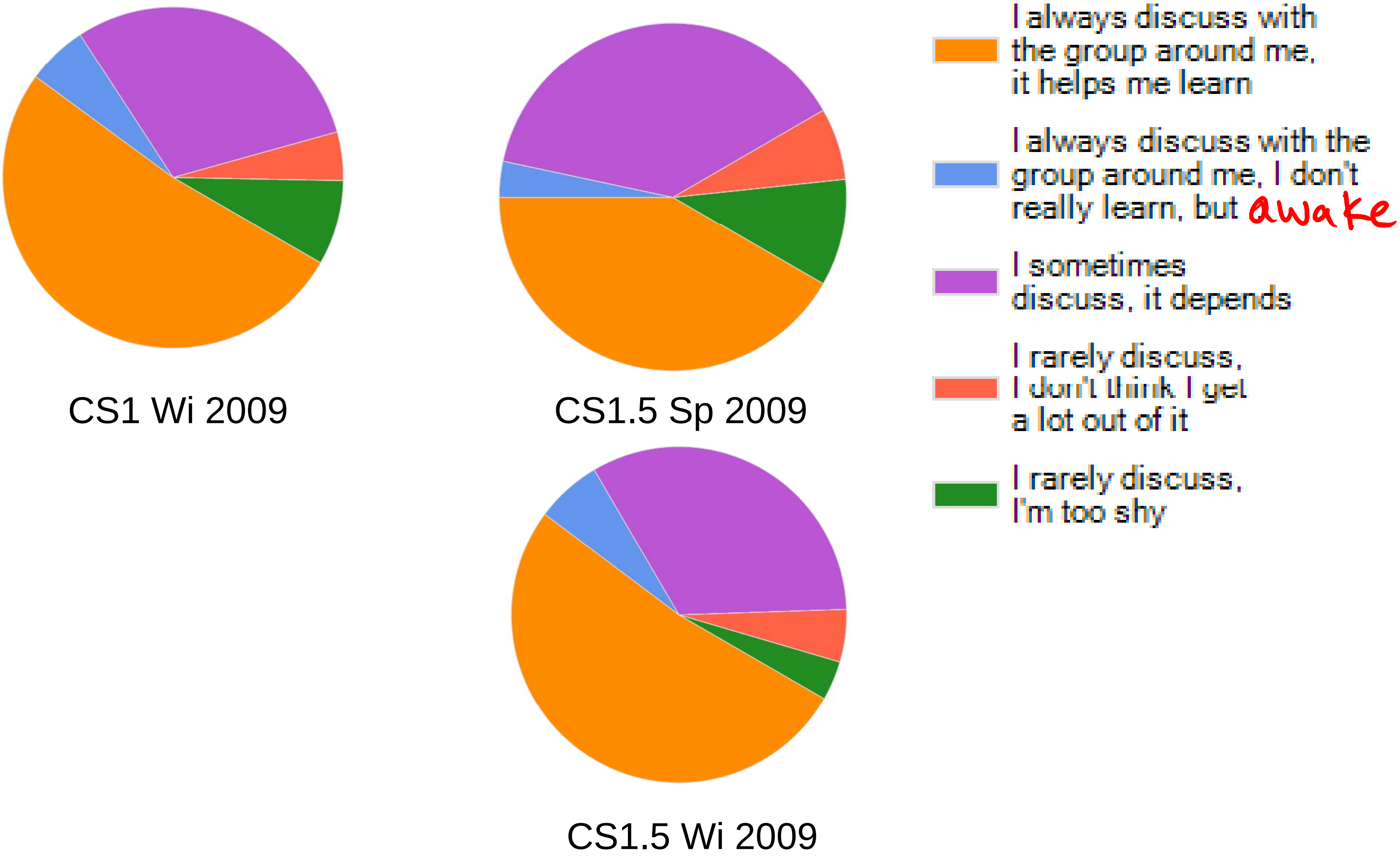
CS1.5 Wi 2009

Please rate your agreement with this statement:

Clickers with discussion is valuable for my learning

Course	Disagree	Neutral	Agree
CS1 Winter	14%	8%	78%
CS1.5 Spring (2nd term)	7%	7%	87%
CS1.5 Winter (2nd term)	8%	11%	81%

Which of the following best describes your discussion practices in the class this term?

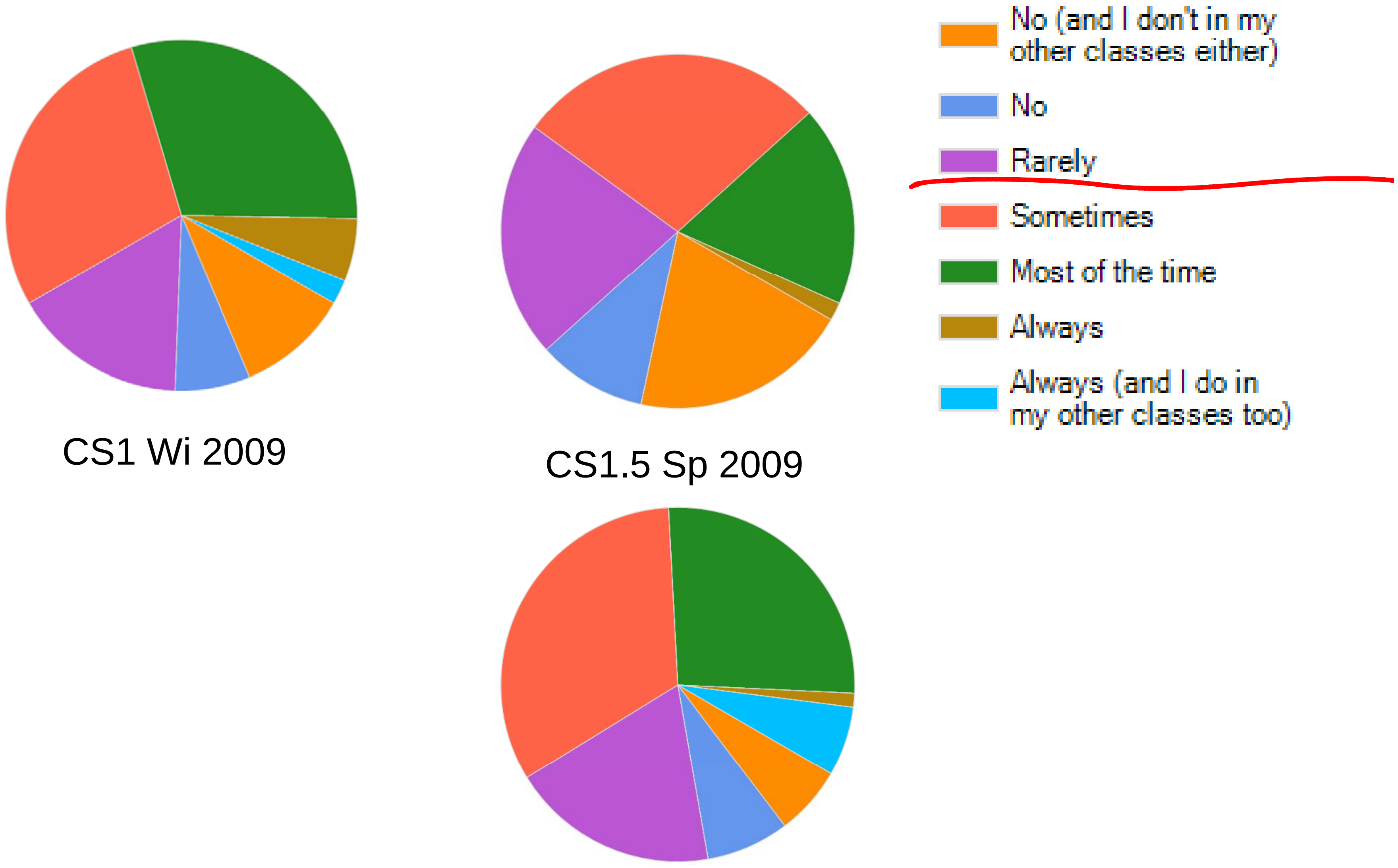


CS1 Wi 2009

CS1.5 Sp 2009

CS1.5 Wi 2009

Did you really read the textbook before coming to class?

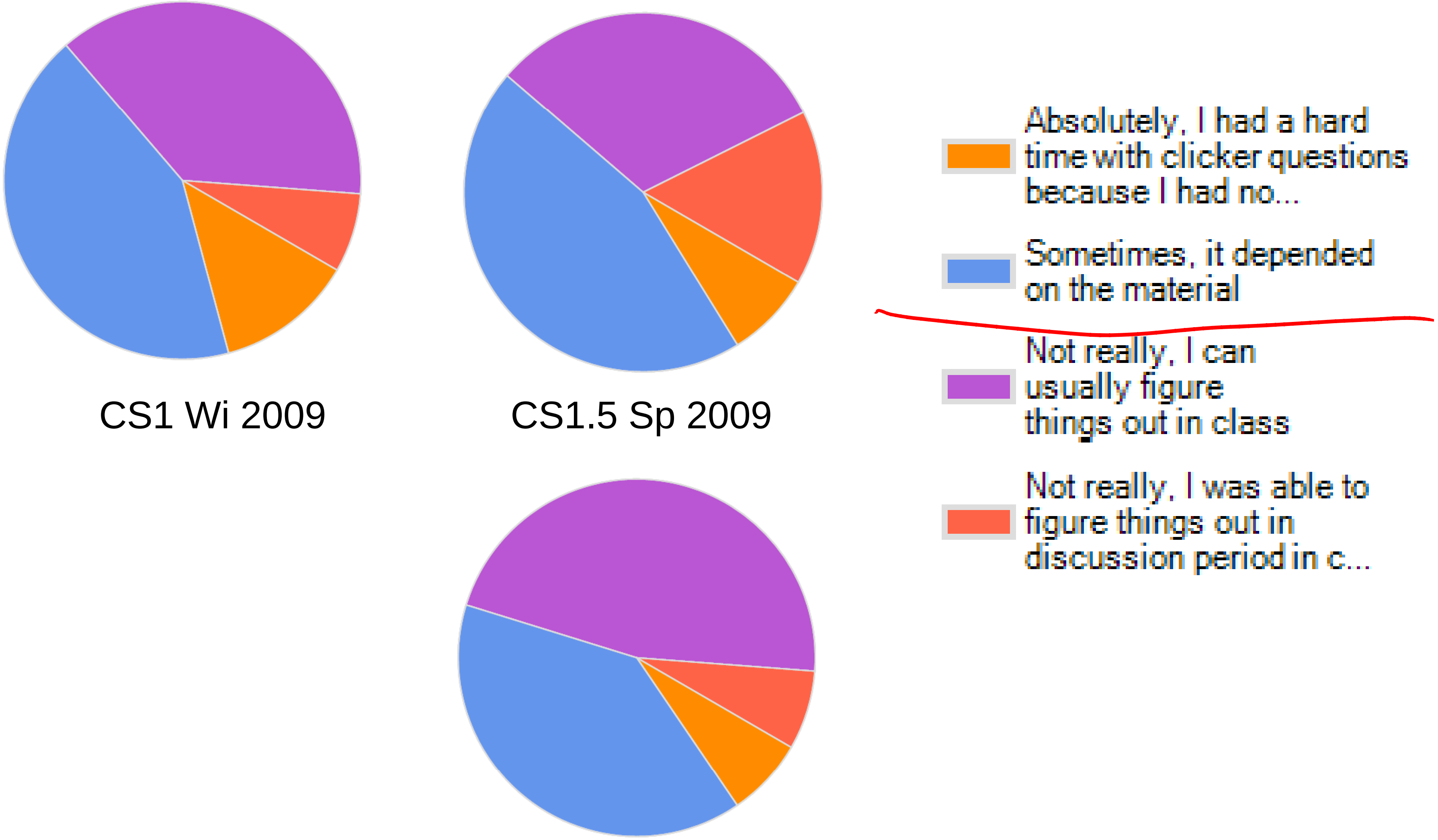


CS1 Wi 2009

CS1.5 Sp 2009

CS1.5 Wi 2009

If you rarely read the textbook, do you think it hindered your learning in class?



CS1 Wi 2009

CS1.5 Sp 2009

CS1.5 Wi 2009

What makes a good clicker question?

- Code/Calculations: (29)
  - Clicker questions that apply to a sample method/piece of code work best.
  - Example code problems, piece by piece, helps to know how to write my own programs (like the recursion questions)
- Exams/Quiz Prep: (19)
  - "ones that appear on exams and quizzes, or are useful to my actual programming experience."
- Tricky (8)
  - The questions that were helpful were the one that I got right, but the majority of the class got wrong. This is because it helped me to understand that I really knew the material for a tricky question. These "tricky" questions helped me get a fuller understanding of the material.

Good quotes

- “The ones I miss (yeah, real useful, I know...). Generally I remember more from the ones I get wrong than the ones I get right because I'll be on a test and be like, "Oh yeah, I made that stupid mistake in class... It should be this instead." It's generally like that with tests, too--if I miss something once, I remember it better and (usually) get it right the next time around.”

What makes a bad clicker question?

- Too Fast (7)
  - I can figure out everything if given enough time. It just takes a moment or two to really digest what the slide says and then answer. Sometimes slides are flashed by too quickly and I just pick something at random, then sort out my answer later.
- Tricky (6)
- Conceptual (5, 6)

Go forth!

- You too can do it!
- Resources:
  - PeerWise DB of clicker questions?
  - Explanations for each of the answers…
  - Voting record
  - Chapter and Page info
- What would you want to know in a video?
  - About CS, different than physics

Midterm 4: simple int param pass: 97% correct

4) (1pt) What is printed when we run this code?

```
 public static void main(String[] args)
{
  double y = 33.3;
  foo(y);
  System.out.println(y);
}

public static void foo(double x)
{
  x = x + 11.1;
  System.out.print(x + “, “);
}
```

A) 33.3, 33.3
B) 33.3, 44.4
C) 44.4, 33.3
D) 44.4, 44.4
E) This code does not compile because the foo method doesn’t have a variable y

Clicker: 1st day class: 3.31.09: Solo: 34% Group: 60%

A. 33
34

B. 34
33

C. 33
33

D. 34
34

E. None
of the
above

```
public static void main (String[] args)
{
  int x = 33;
  foo(x);
  S.o.pln(x);
}

public static void foo (int x)
{
  x++;
  S.o.pln(x);
}
```

Backup Midterm Exam 5: “more complex” int param pass:
95% on int pass only (not return: A or C)

5) (2pts) What is printed when we run this code?

```
 public static void main(String[] args)
{
  double y = 33.3;
  double x = 99.9;
  x = foo(y,x);
  System.out.println(y +”, “+ x);
}

public double void foo(double y, double x)
{
  y = y + 11.1;
  x = x + 0.1;
  return x;
}
```

A: 33.3, 99.9
B: 44.4, 99.9
C: 33.3, 100.0
D: 44.4, 100.0
E: None of the above

Midterm 6: Object param pass (complex with ordering): B or C: 97%

```
public static void main(String[] args)
  {
    //The Car constructor takes car color, and number of gallons of gas in tank
    Car beths = new Car (“red”, 0 );
    Car lisas = new Car(“silver”, 0);
    foo(beths, lisas);
    System.out.println("Beth: " + beths.getGas() +
                        " Lisa: " + lisas.getGas());
  }

  public static void foo(Car lisas, Car beths)
  {
    beths.fill(5.5); //Assume adds to gallons of gas stored
    lisas.fill(2.2); //Assume adds to gallons of has stored
  }
```

A) Beth: 0.0 Lisa: 0.0
B) Beth: 5.5 Lisa: 2.2
C) Beth: 2.2 Lisa: 5.5
D) Beth: 7.7 Lisa: 7.7
E) None of the above

Clicker: 2nd day (guest lecturer)
Solo: 56% Group: 49%

A. 33
   34

B. 34
   33

C. 33
   33

D. 34
   34

E. None of the above

```
public static void main (String[] args)
{
  //Assume 33 is how balance already spent
  CreditCard visa = new CreditCard(33);
  foo(visa);
  S.o.p(visa.getBalance());
}

public static void foo(CreditCard visa)
{
  int temp = visa.getBalance();
  visa.setBalance(temp+1);
  S.o.p(visa.getBalance());
}
```

Midterm 7: new and return
new only: A or B: 83%

7. (2pts) What is printed when we run this code?

```
public static void main(String[] args)
  {
    //The Car constructor takes car color, and number of gallons of gas in tank
    Car beths = new Car (“red”, 0);
    Car otherCar;
    otherCar = foo(beths);
    System.out.println("Beth: " + beths.getGas() +
                       " Other: " + otherCar.getGas());
  }

  public static Car foo(Car beths)
  {
    beths = new Car("blue”, 0);
    beths.fill(12.0); //assume adds to gallons of gas stored
    return beths;
  }
```

A) Beth: 0.0 Other: 0.0
B) Beth: 0.0 Other: 12.0
C) Beth: 12.0 Other: 0.0
D) Beth: 12.0 Other: 12.0
E) None of the above

Clicker: split lecture 4.02/4.07
Solo: 53% Last Group: 74%

A. 33
   34

B. 34
   33

C. 33
   33

D. 34
   34

E. None of the above

```
public static void main (String[] args)
{
  //Assume 33 is how balance already spent
  CreditCard visa = new CreditCard(33);
  foo(visa);
  S.o.p(visa.getBalance());
}

public static void foo(CreditCard visa)
{
  visa = new CreditCard(50);
  S.o.p(visa.getBalance());
}
```

Midterm 15: direct copy: 84%
Clicker: Solo 24% Group 23%

15) (2pts) What is output by this code?

```
ArrayList foo = new ArrayList();
for (int i = 0; i < 10; i++)
{
  Integer bar = new Integer(i);
  foo.add(bar);
}
for (int j = 0; j < 5; j++)
{
  foo.remove(j*2);
  System.out.println(foo.size()+”,“);
}
```

10,8,6,4,2

9,7,5,3,1

10,10,10,10,10

9,9,9,9,9

None of the above.

Midterm 16: ordering of exception catch: 59% (exam added option D) Solo: 19% Group: 26%

16) (1pt) What is printed when the file input.txt does not exist?

```
String line = null;
try {
 BufferedReader reader =
    new BufferedReader(new FileReader(“input.txt”));
 line = reader.readLine();
 reader.close();
}
catch (Exception ex)
{ S.o.p(“Error1”; System.exit(0); }
catch (FileNotFoundException ex)
{ S.o.p(“Error2”; System.exit(0);}
```

A. Error1

B. Error2

C. Error1Error2

D. This code does not compile since, in the catch blocks, both Exceptions are named ex

E. None of the above

Midterm 17: array underpinning of ArrayList - remove
77% C (84% A or C)

17. (2pts) What indexing pattern do we want for our assignment to “remove” the element at index 4 in an array called foo (which has 7 values in it)?
(mimicking what ArrayList does in its call to remove)

```
A) foo[3] = foo[4]
   foo[4] = foo[5]
   foo[5] = foo[6]
B) foo[4] = foo[3]
   foo[5] = foo[4]
   foo[6] = foo[5]
C) foo[4] = foo[5]
   foo[5] = foo[6]
D) foo[5] = foo[4]
   foo[6] = foo[5]
E) None of the above
```

Clicker: array underpinning of ArrayList - add
Solo: 23% Group: 60%

10	20	30	40	50	60	70	

What indexing pattern do we want for our assignment to "move elements up" to insert at 4

A. foo[3] = foo[4]
   foo[4] = foo[5]
   foo[5] = foo[6]

B. foo[5] = foo[4]
   foo[6] = foo[5]
   foo[7] = foo[6]

C. foo[7] = foo[6]
   foo[6] = foo[5]
   foo[5] = foo[4]

D. foo[6] = foo[5]
   foo[5] = foo[4]
   foo[4] = foo[3]

E. None of the above