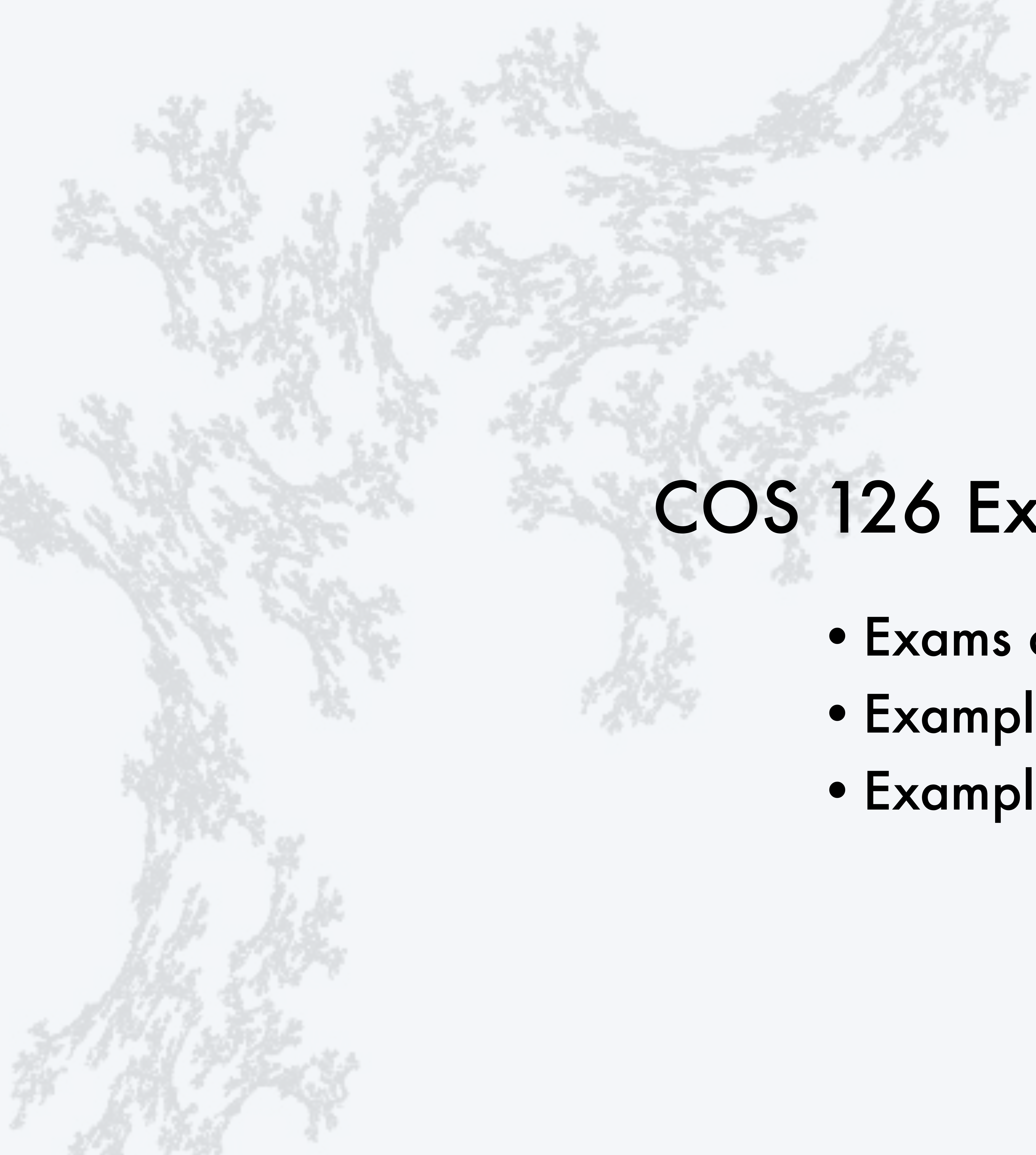


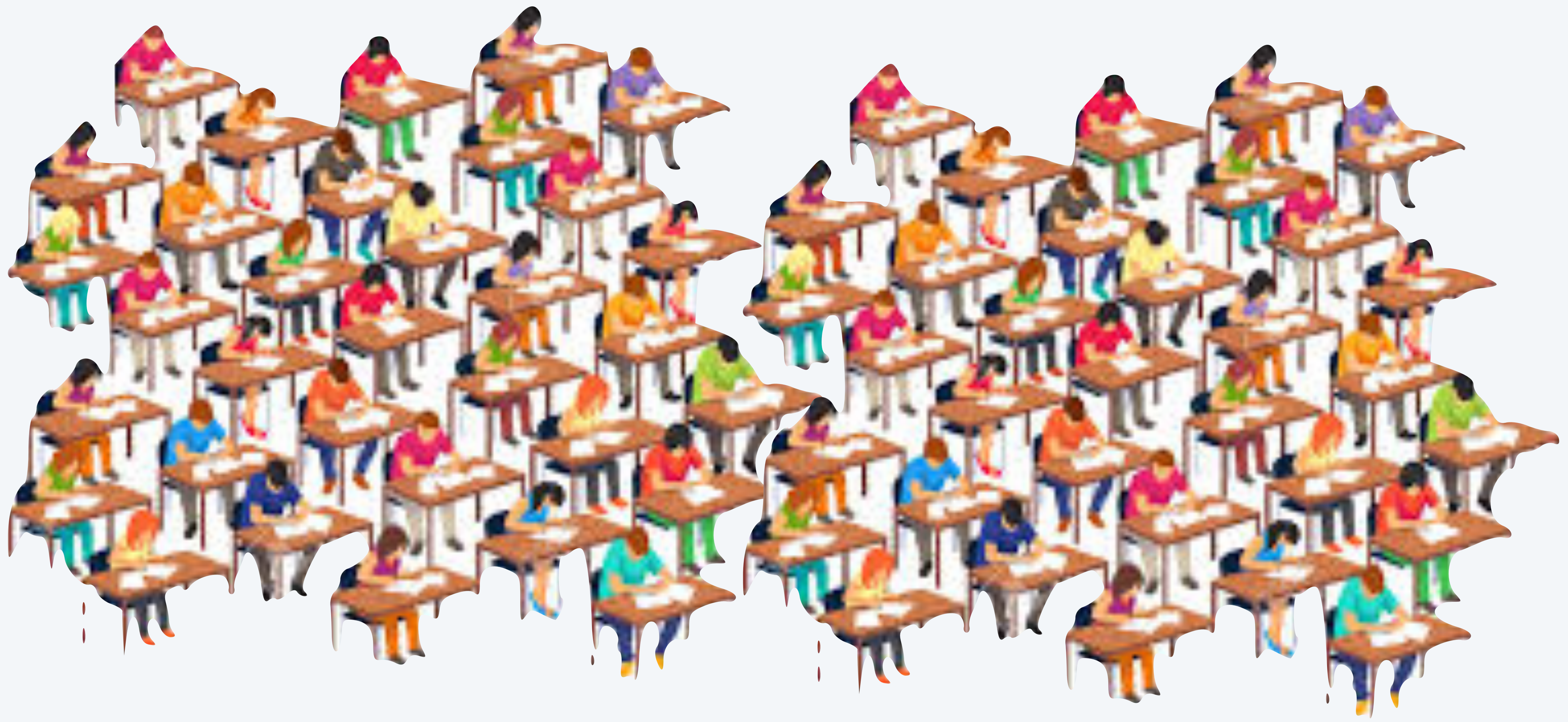
COS 126 Exam Review

- Exams overview
- Example programming exam
- Example written exam questions (part 1)

Exams overview (revisited)

We have exams in the fall

- Two written exams.
- Two programming exams.
- Prep sessions in class meetings
- **No exam "midterm week"**
- **No final exam**



Programming exams.

- October 11 and December 6.
- Mini in-class assignments.
- *“Can you write a short program?”*
- You will need to practice.

Written exams.

- October 18 and December 13.
- Written by RS. $\longleftarrow$ *Note: RS has been known to re-use questions*
- *“Did you watch the lectures and do the reading?”*
- One question per lecture (roughly).

Exams tab on the booksite

See Exams tab for full details and old exams.

- Read carefully *before* each exam.
- Policies are the contract between us and you.

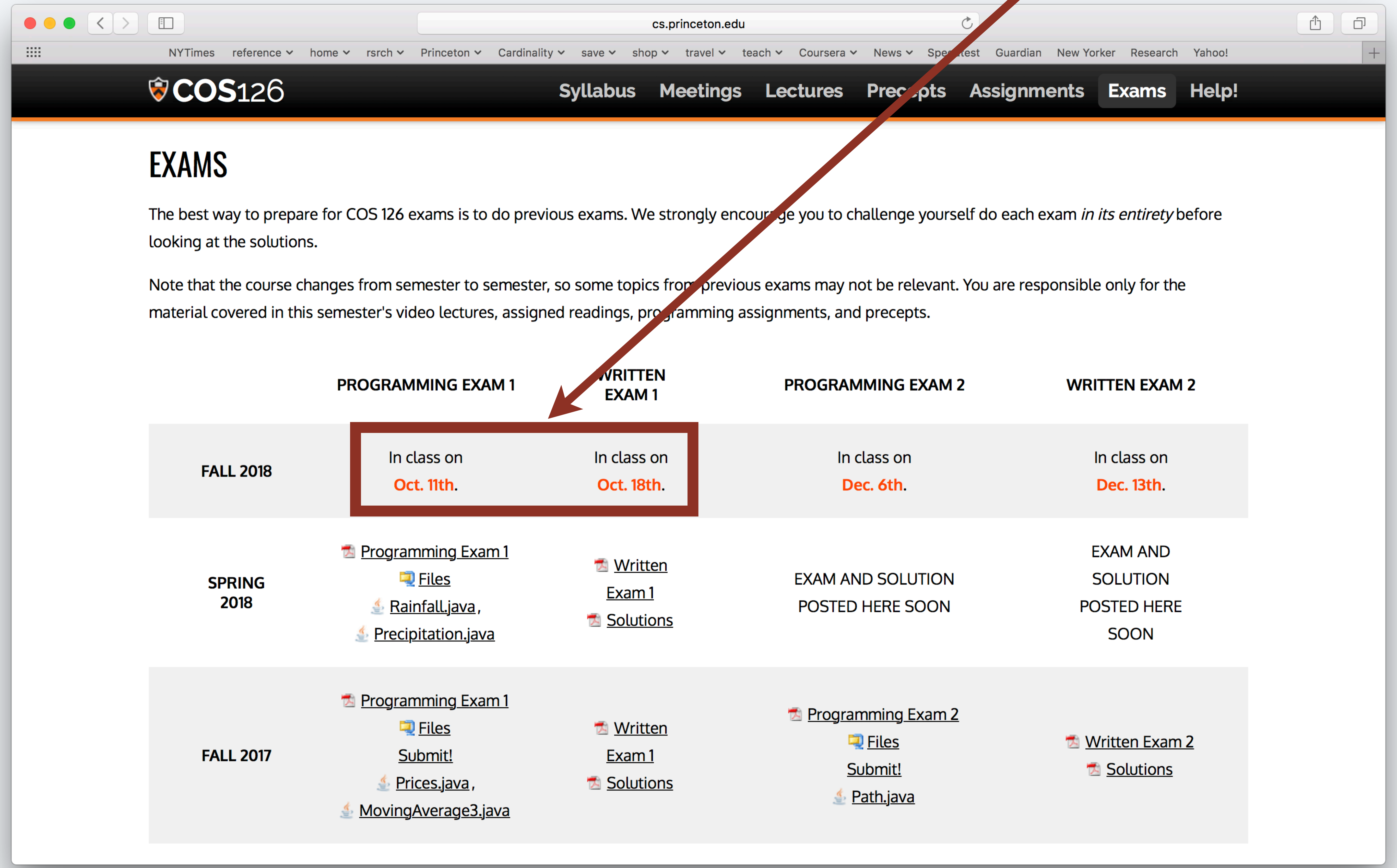
Policies (programming exam).

- Open course materials.
- No other web access.
- No outside communication.

Policies (written exam).

- Closed book/notes/computer.
- 1 page (two sides) cheatsheet.

Watch this space for details


cs.princeton.edu
NYTimes reference home rsrch Princeton Cardinality save shop travel teach Coursera News Speedtest Guardian New Yorker Research Yahoo!
COS126
Syllabus Meetings Lectures Precepts Assignments Exams Help!

EXAMS

The best way to prepare for COS 126 exams is to do previous exams. We strongly encourage you to challenge yourself do each exam *in its entirety* before looking at the solutions.

Note that the course changes from semester to semester, so some topics from previous exams may not be relevant. You are responsible only for the material covered in this semester's video lectures, assigned readings, programming assignments, and precepts.

	PROGRAMMING EXAM 1	WRITTEN EXAM 1	PROGRAMMING EXAM 2	WRITTEN EXAM 2
FALL 2018	In class on Oct. 11th.	In class on Oct. 18th.	In class on Dec. 6th.	In class on Dec. 13th.
SPRING 2018	Programming Exam 1 Files Rainfall.java, Precipitation.java	Written Exam 1 Solutions	EXAM AND SOLUTION POSTED HERE SOON	EXAM AND SOLUTION POSTED HERE SOON
FALL 2017	Programming Exam 1 Files Submit! Prices.java, MovingAverage3.java	Written Exam 1 Solutions	Programming Exam 2 Files Submit! Path.java	Written Exam 2 Solutions

Things to remember about inclass exams

We know that you don't have much time.

- Exams are 50 minutes.
- "One page" programming exams.
- Five-minute questions on written exams.

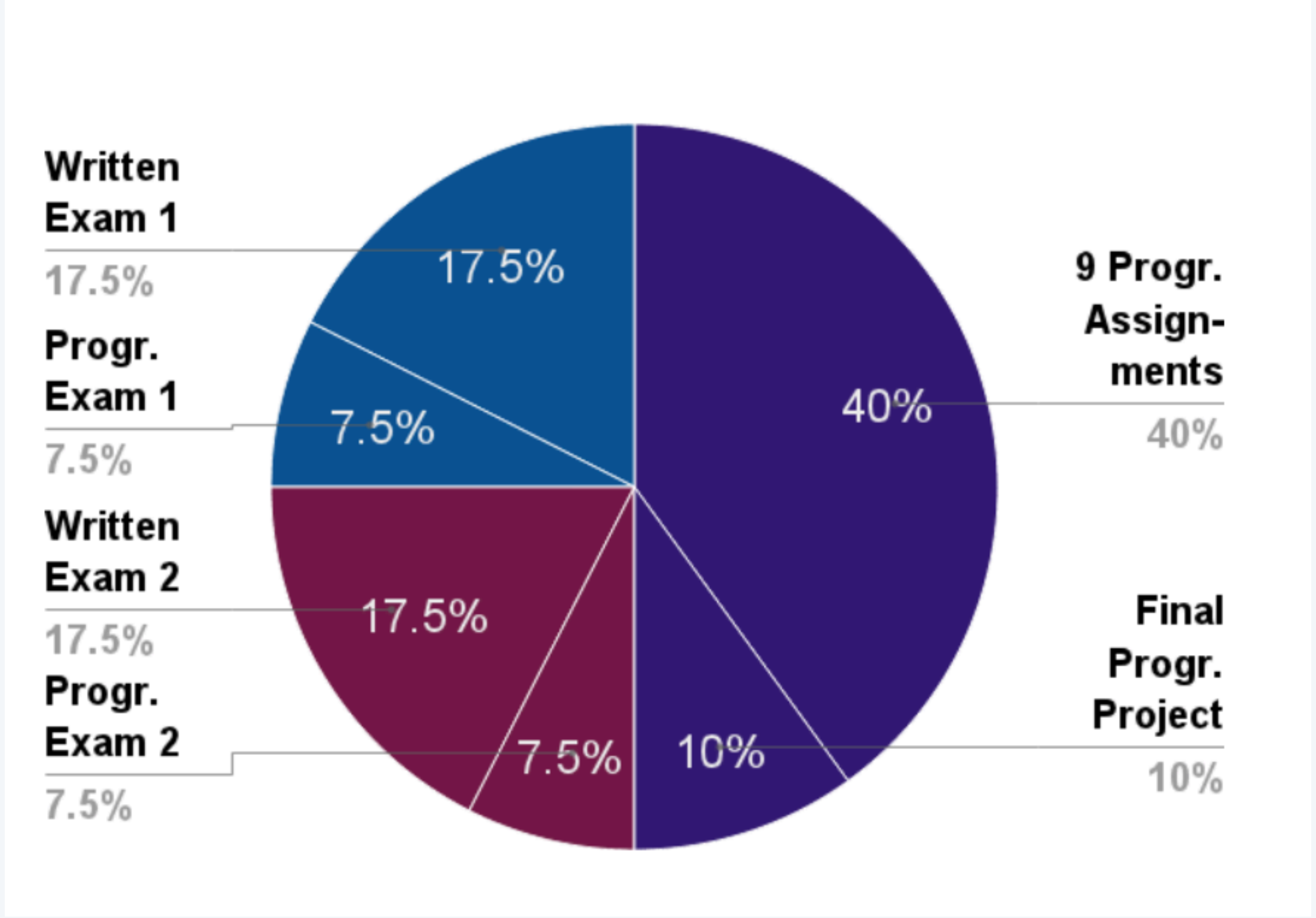
We have to grade the exams.

- 400+ exams.
- No open-ended questions.
- Fully prepared rubrics.

Old exams are not completely reliable.

- Course offerings differ slightly.
- We have made mistakes in the past.

Exams are only part of the story.



Programming Exam Logistics

Writing a short program in 50 minutes can be a challenge for anyone.

- You will use your own computer.
- You will download and edit a template.
- You will submit your solution in the same way as you do for assignments.

You don't all fit in this room.

- Pay attention and know where to go.
- Arrive early.
- Make sure your computer is charged.

Advice: Practice, practice, practice.

- Write some short programs on your own.
- Attend the practice programming exam.
- Try a past programming exam (untimed).
- Try another one (timed).

Programming Exam 1 Part 1

Q. *Can you write a simple program on your own ?*

Example (Fall 2015).

Part 1. Write programs that find the number of distinct values among the integers on standard input, assuming that the input is nonempty and in sorted order.

Your task. Add code to the file `Count1.java` to print the number of integers on standard input and the number of distinct values among those integers.

Details. Write a single loop that uses `StdIn.readInt()` to read each integer one at a time, but *do not save them in an array*. To compute the number of distinct values, add code to the loop to update `distinct` if the new value just read differs from the value read just before it.

Exams Info ***gave instructions to load this before the exam***

```
public class Count1
{
  public static void main(String[] args)
  {
     int count = 1;
     int distinct = 1;
     // YOUR CODE HERE
  }
}
```

you also get a test file and desired output

```
% more testCount1tiny.txt
1 1 1 1 2 2 2 2 4 4 4 4 5 5 6 6 9 9

% java Count1 < testCount1tiny.txt
6 distinct values among 18 integers
```

Programming Exam Strategy

Write and submit the easiest code before tackling the hard part.

Your task. Add code to *print the number of integers on standard input* and the number of distinct values among those integers.

```
% more testCount1tiny.txt
1 1 1 1 2 2 2 2 4 4 4 4 5 5 6 6 9 9

% java Count1 < testCount1tiny.txt
1 distinct values among 18 integers
```

```
% java Count1 < testCount1tiny.txt
6 distinct values among 18 integers
```

```
public class Count1
{
   public static void main(String[] args)
   {
      int count = 1;                  // number of integers
      int distinct = 1;      // number of different ones
      int val = StdIn.readInt();
      while (!StdIn.isEmpty())
      {
         int newVal = StdIn.readInt();
         count++;           // count the integers
         if (newVal != val)
         {
            distinct++;     // count the different ones
            val = newVal;
         }
      }
      StdOut.println(distinct +
              " distinct values among "
                         + count + " integers");

   }
}
```

Programming Exam 1 Part 2

Q. *Can you quickly apply something you have recently learned ?*

Example (Fall 2015).

Part 2. Assume that the integers on standard input are nonnegative and less than a value M given as the first command-line argument, *but not necessarily in order*.

Details. To compute the needed values, use two loops. First, write a loop that reads the integers one at a time from standard input, counts them, and uses the boolean array b[] to record which values have been seen: when you read a value val, set b[val] to true. Second, write a loop that counts the number of true values in b[] (the number of distinct values in the input).

Q. *Do you understand "coupon collector" ?*

***Exams Info** gave instructions to load this before the exam*

```
public class Count2
{
   public static void main(String[] args)
   {
      int M = Integer.parseInt(args[0]);
      boolean[] b = new boolean[M];
      // YOUR CODE HERE
   }
}
```

the description includes an example

4 1 4 1 5 9 2 6 2 4 1 4 1 5 9 2 6 2

i	0	1	2	3	4	5	6	7	8	9
b[i]	F	T	T	F	T	T	T	F	F	T

Programming Exam 1 Part 2 Fall 2015 Solution

```
public class Count2
{
    public static void main(String[] args)
    {
        int M = Integer.parseInt(args[0]);  // no number higher than M - 1
        boolean[] b = new boolean[M];       // true means index was input

        // loop to read input, count integers and fill boolean array
        int count = 0;
        while (!StdIn.isEmpty())
        {
            int val = StdIn.readInt();
            b[val] = true;
            count++;
        }

        // count distinct numbers
        int distinct = 0;
        for (int i = 0; i < M; i++)
            if (b[i]) distinct++;

        // output
        StdOut.println(distinct + " distinct values among " + count + " integers");
    }
}
```

Written Exam Logistics

The first exam is on Thursday Oct. 18.

- Prep session (first half) later today.
- Prep session (second half) Tuesday Oct. 16.

You don't all fit in this room.

- Pay attention and know where to go.
- Arrive early.
- No calculator/phone/computer/headphones

Advice.

- Review lectures/reading.
- Try an old exam (untimed).
- Try another one (timed).
- Review a few more.

Example question: Basics

Q. *Do you understand types and Java's type conversion and precedence rules ?*

Ex. (Fall 2014 Question 1) Give the type and value of each of the following Java expressions, supposing that it is used as the argument of a `println()` call.

`(3 < 2) && (1 > 0)`	boolean	FALSE
`"800" + 99`	String	80099
`800 + 99 + "A"`	String	899A
`3 + (int) Math.random()`	int	3
`(double)(3 / 2)`	double	1.0
`3 / 2.0 + 2 / 5`	double	1.5
`(8 <= 2) || (2e88 <= 88e2)`	boolean	FALSE
`Integer.parseInt("8.5*2")`	ILLEGAL	

Example question: Arrays

Q. *Do you understand basic properties and rules about Java arrays ?*

Ex. (Fall 2016 Question 2) Which of the following statements are true for Java arrays? Mark each statement as either TRUE or FALSE.

An array can't simultaneously store both an element of type `double` and an element of type `boolean`.	TRUE
Once you create an array, you cannot change its type.	TRUE
You must access the elements in an array in sequential order, e.g., you cannot access a[5] until you have accessed a[0], a[1], through a[4].	FALSE
If `a[]` is a boolean array of length 126, then the expression `a[i]` will evaluate arbitrarily to either true or false if the index `i` is equal to 126.	FALSE
If `a[]` and `b[]` are two arrays of length 2, then `a == b` is `true` if and only if both `a[0] == b[0]` and `a[1] == b[1]` are `true`.	FALSE

Example question: Loops and conditionals

Q. *Can you figure out the effect of a simple Java program that uses* `while` *and* `if` *statements?*

Ex. (Fall 2014 Question 2) Fill in the trace for *just after* each iteration of the outer `for` loop in this program:

```
int[] a = new int[N];
a[0] = 1;
for (int i = 1; i < N; i++)
{
   int sum = 0;
   for (int j = 0; j < i; j++)
      sum = sum + a[j];
   a[i] = 1 + (2 * sum) / i;
}
```

i	sum	a[i]
0		1
1	1	3
2	**4**	**5**
3	**9**	**7**
4	**16**	**9**
5	25	11
6	36	13

Write *one line of code* that could replace the body of the outer loop.

`a[i] = 2*i + 1;`

Example question: Input and output

Q. *Do you understand basic ways of communicating with your programs* ?

Ex. (S2011 Q4) Give the results of invoking this program with the given commands.

```
public class Q4
{
    public static void main(String[] args)
    {
        int curr = StdIn.readInt();
        StdOut.print(curr + " ");
        int prev = curr;
        while (!StdIn.isEmpty())
        {
            curr = StdIn.readInt();
            StdOut.print((prev + curr) / 2 + " ");
            prev = curr;
        }
        StdOut.println();
    }
}
```

```
% more input.txt
2 4 6 8 10 12 8 2
% java Q4 < input.txt
```

```
2 3 5 7 9 11 10 5
```

```
% java Q4 < input.txt | java Q4
```

```
2 2 4 6 8 10 10 7
```

Note: It prints the first number, then the average of each number and its predecessor.

Mark your calendar

***Tuesday* October 9: Practice programming exam**

***Thursday* October 11: PROGRAMMING EXAM**

***Tuesday* October 16: Written exam prep (part 2)**

***Thursday* October 18: WRITTEN EXAM**