

**Details, Details**

*Create a class named Slideshow and  
save it to a file named Slideshow.java.*

```
public class Slideshow  
{  
}
```

*A Slideshow keeps track of  
an array of Pictures and an array of Sounds.*

```
public class Slideshow
{
    private Picture[] images;
    private Sound[] sounds;
}
```

*The class should have a constructor that lets users pass two arrays when creating a new Slideshow object.*

```
public class Slideshow
{
    private Picture[] images;
    private Sound[] sounds;

    public Slideshow(Picture[] see, Sound[] hear)
    { ... }
}
```

*Slideshow should have a method named show( )  
that plays the slideshow.*

```
public class Slideshow
{
    private Picture[] images;
    private Sound[] sounds;

    public Slideshow(Picture[] see, Sound[] hear)
    { ... }

    public show()
    { ... }
}
```

*Your class should have a `main( )` method  
that creates a slideshow and shows it.*

```
public class Slideshow
{
    ...

    public static void main( String[] args)
    {
        // code to make two arrays ...
        Slideshow presentation = new Slideshow(...);
        presentation.show();
    }
}
```

# The Premise ...

Last time, we simulated a PowerBall entry using our `Die` class.

So I gave myself an exercise to do:

*Write a `PowerBallEntry` class that uses dice to initialize the objects.*

(The `main()` method we wrote for `Die` gave me just what I needed!)

## ... The Exercise

How can we tell if someone's `PowerBallEntry` is a winner?

*Write a `PowerBallEntry` method named `matches()` that compares one entry to another.*

```
public boolean matches( PowerBallEntry another )
```

## ... The Rest of the Exercise

Winning PowerBall is easy.  
Right?

*Let's write a `main()` method that  
keeps drawing entries until it finds  
an entry that matches the first.*

*At the end, display how many tickets were drawn.*

# How to Write to a Text File

A simple way of writing a file is:

1. Create a file object.
2. Write data to it using its `write()` and `newLine()` methods.
3. Close the file.

#2 and #3 are just like what we've done in the past.  
#1 requires a "*trust me*" moment or two — for now.

# Comma-Separated Values

A typical entry from my class list to start the term:

```
"810061 04,SCHAFER,J,BEN,123456,schafer@cs.uni.edu"
```

where the fields are:

```
"Course,Last Name,First Name,Middle Name,UNI ID,e-mail"
```

# Winning Powerball

The winner is #37919215

The winner is #276263480

The winner is #81388798

The winner is #132464297

# Quick Exercise

*Write a Student method named emailDomain()  
that returns just the primary domain  
of the student's e-mail address.*

```
> Student s = new Student( "810061 04,SCHAFER,J,BEN,123456,schafer@cs.uni.edu" );
> s.emailDomain()
"uni.edu"

> Student t = new Student(
    "810061 04,GUY,MADE,UP,234567,indecipherable.mess@gmail.com" );
> t.emailDomain()
"gmail.com"
```

# How to Read to a Text File

A simple way of writing a file is:

1. Create a file object.
2. Read data from it using its `readLine()` method.
3. Close the file.

#2 and #3 are just like what we've done in the past.  
#1 requires a "*trust me*" moment or two — for now.