Input from Files
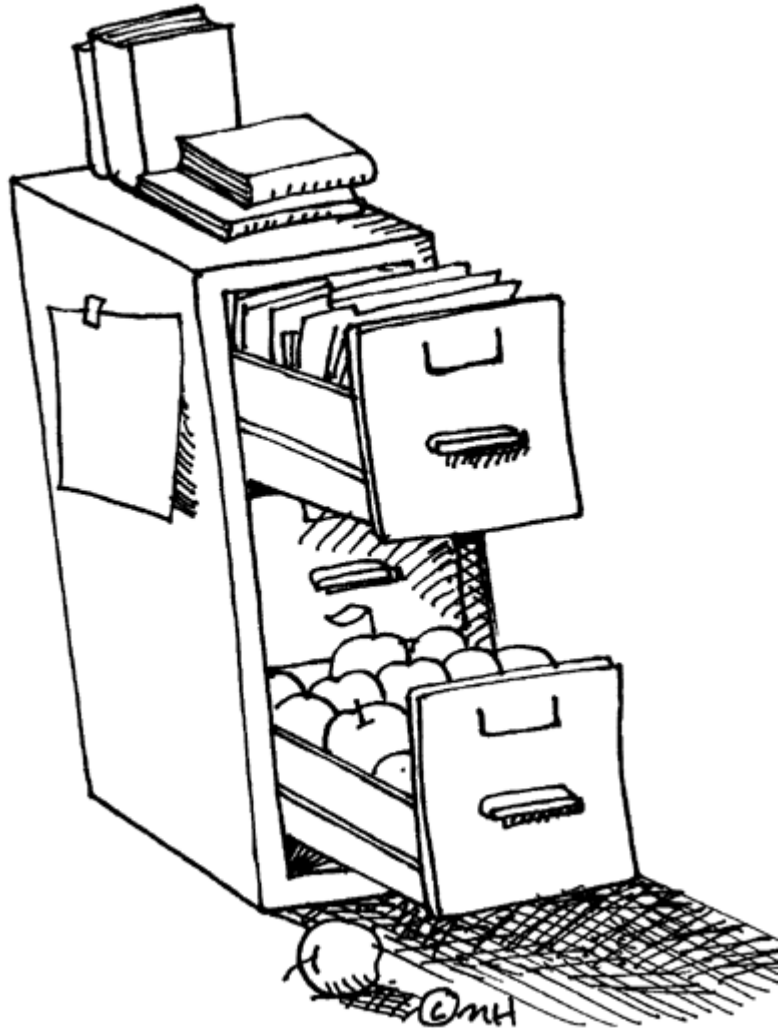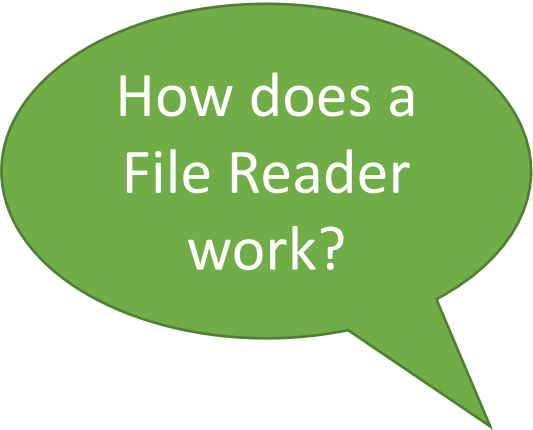
Buffered Reader

- Input from files is always text.
- You can convert it to ints using Integer.parseInt()

- We use BufferedReaders to minimize the number of reads to the file.
- The Buffer reads a whole chunk of information from the file and hands it to us one piece at a time.
- That way, each read operation is faster.

This is a file.
It has text in it.
The text can be read.
Line by line.
The program can …
… use the data.
Like open a saved file.
Maybe a high scores list.
Or some user settings.

How does a File Reader work?

```
in.readLine();
```

This is a file.

This is a file.
It has text in it.
The text can be read.
Line by line.
The program can …
… use the data.
Like open a saved file.
Maybe a high scores list.
Or some user settings.

```
in.readLine();
```

This is a file.
It has text in it.
The text can be read.
Line by line.
The program can …
… use the data.
Like open a saved file.
Maybe a high scores list.
Or some user settings.

It has text in it.

`in.readLine();`

This is a file.
It has text in it.
The text can be read.
Line by line.
The program can …
… use the data.
Like open a saved file.
Maybe a high scores list.
Or some user settings.

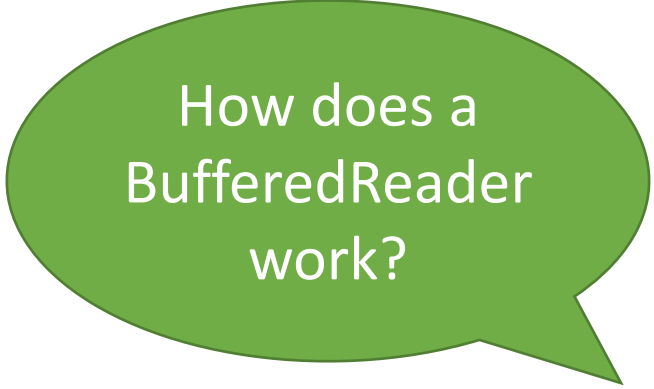The text can be read.

This is a file.
It has text in it.
The text can be read.
Line by line.
The program can …
… use the data.
Like open a saved file.
Maybe a high scores list.
Or some user settings.

How does a BufferedReader work?

```
in.readLine();
```

This is a file.
It has text in it.
The text can be read.
Line by line.
The program can …
… use the data.
Like open a saved file.
Maybe a high scores list.
Or some user settings.

This is a file.
It has text in it.
The text can be read.
Line by line.

This is a file.

`in.readLine();`

This is a file.
It has text in it.
The text can be read.
Line by line.
The program can …
… use the data.
Like open a saved file.
Maybe a high scores list.
Or some user settings.

This is a file.
It has text in it.
The text can be read.
Line by line.

It has text in it.

`in.readLine();`

This is a file.
It has text in it.
The text can be read.
Line by line.
The program can …
… use the data.
Like open a saved file.
Maybe a high scores list.
Or some user settings.

This is a file.
It has text in it.
The text can be read.
Line by line.

The text can be read.

```java
import java.io.*;
public class readIn {
  public static void main (String args[]) {
     new readIn ();
  }
  public readIn () {
     BufferedReader in;
     try {
       in = new BufferedReader (new FileReader ("numbers.txt"));
       String next = in.readLine ();
       in.close ();
       System.out.println (total);
     }
     catch (IOException e) {
       System.out.println ("Error opening file " + e);
     }
   }
}
```

```java
import java.io.*;
public class readIn {
    public static void main (String args[]) {
        new readIn ();
    }
    public readIn () {
        BufferedReader in;
        try {
            in = new BufferedReader (new FileReader ("numbers.txt"));
            String next = in.readLine ();
            in.close ();
            System.out.println (total);
        }
        catch (IOException e) {
            System.out.println ("Error opening file " + e);
        }
    }
}
```

Import the right library

```java
import java.io.*;
public class readIn {
   public static void main (St
      new readIn ();
   }
   public readIn () {
      BufferedReader in;
      try {
         in = new BufferedReader (new FileReader ("numbers.txt"));
         String next = in.readLine ();
         in.close ();
         System.out.println (total);
      }
      catch (IOException e) {
         System.out.println ("Error opening file " + e);
      }
   }
}
```

Import the right library

Open a Buffered Reader.
Sort of a 'scanner' for files.

```java
import java.io.*;
public class readIn {
  public static void main (St:
    new readIn ();
  }
  public readIn () {
    BufferedReader in;
    try {
      in = new BufferedReader (new FileReader ("numbers.txt"));
      String next = in.readLine ();
      in.close ();
      System.out.println (total);
    }
    catch (IOException e) {
      System.out.println ("Error opening file " + e);
    }
  }
}
```

Import the right library

Open a Buffered Reader.
Sort of a 'scanner' for files.

Send in the file name.

```java
import java.io.*;
public class readIn {
    public static void main (St
        new readIn ();
    }
    public readIn () {
        BufferedReader in;
        try {
            in = new BufferedReader (new FileReader ("numbers.txt"));
            String next = in.readLine ();
            in.close ();
            System.out.println (total);
        }
        catch (IOException e) {
            System.out.println ("Error opening file " + e);
        }
    }
}
```

Import the right library

Open a Buffered Reader.
Sort of a 'scanner' for files.

Send in the file name.

Read in a single line.

```java
import java.io.*;
public class readIn {
  public static void main (St
    new readIn ();
  }
  public readIn () {
    BufferedReader in;
    try {
      in = new BufferedReader (new FileReader ("numbers.txt"));
      String next = in.readLine ();
      in.close ();
      System.out.println (total);
    }
    catch (IOException e) {
      System.out.println ("Error opening file " + e);
    }
  }
}
```

Import the right library

Open a Buffered Reader.
Sort of a 'scanner' for files.

Send in the file name.

Read in a single line.

Close the file.

```java
import java.io.*;
public class readIn {
    public static void main (String args[]) {
        new readIn ();
    }
    public readIn () {
        BufferedReader in;
        try {
            in = new BufferedReader (new FileReader ("numbers.txt"));
            String next = in.readLine ();
            in.close ();
            System.out.println (total);
        }
        catch (IOException e) {
            System.out.println ("Error opening file " + e);
        }
    }
}
```

Open the try.

Close the catch.

```java
import java.io.*;
public class readIn {
  public static void main (String args[]) {
    new readIn ();
  }
  public readIn () {
    BufferedReader in;
    try {
      in = new BufferedReader (new FileReader ("numbers.txt"));
      int total = 0;
      String next = in.readLine ();
      while (next != null) {
          total += Integer.parseInt (next);
          next = in.readLine ();
      }
      System.out.println (total);
    }
    catch (IOException e) {
      System.out.println ("Error opening file " + e);
    }
  }
}
```

Looping until done.

Converting a line to an integer