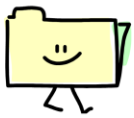
FILES



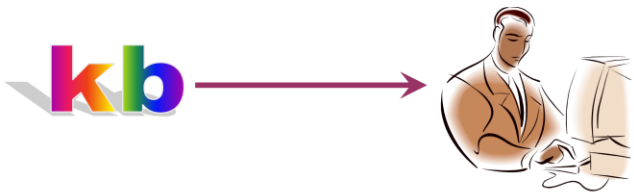
© A+ Computer Science - www.apluscompsci.com

Keyboard Scanner

```
Scanner kb = new Scanner(System.in);

int val  = kb.nextInt();
```

kb connects to the keyboard



© A+ Computer Science - www.apluscompsci.com

kb stores the location/memory address of a Scanner Object that was constructed to read from System.in(the keyboard). `kb.nextInt()` calls the Scanner `nextInt()` method to read in the next integer value.

What is a data file?

A data file is a sequential list of data.

nums.dat
123
567
345
128

© A+ Computer Science - www.apluscompsci.com

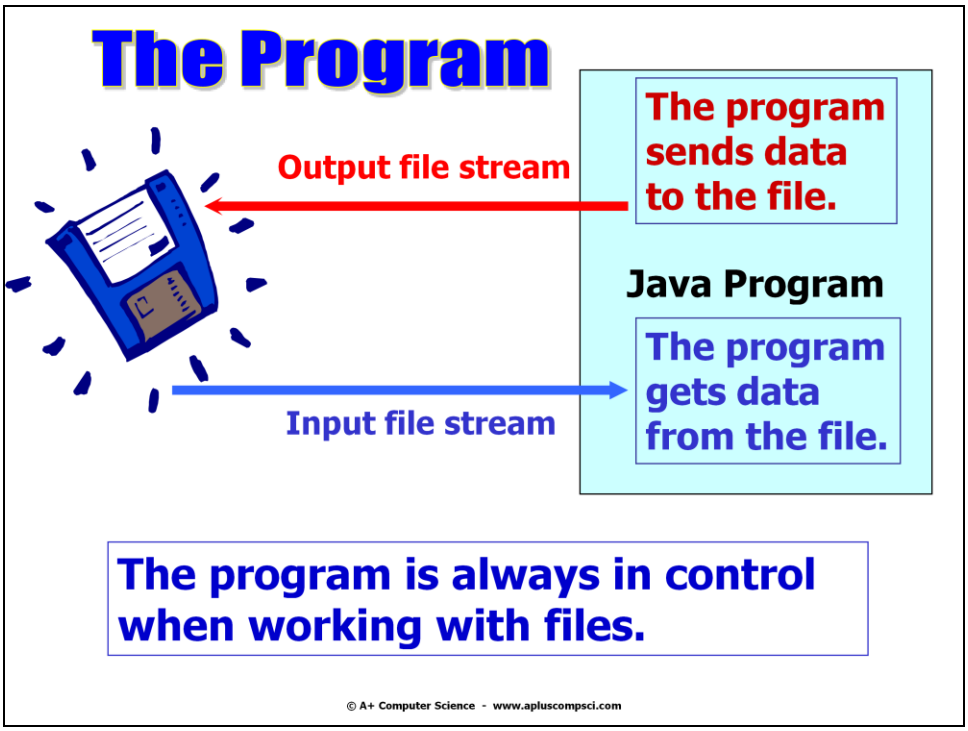
A data file is a sequential, one item right after another, list of values.

All values in a data file can be placed on the same line or on separate lines.

```
10 11 12 14 1 5 2 5 6 3 2 6
```

Or

```
12
14
6
7
3
9
```

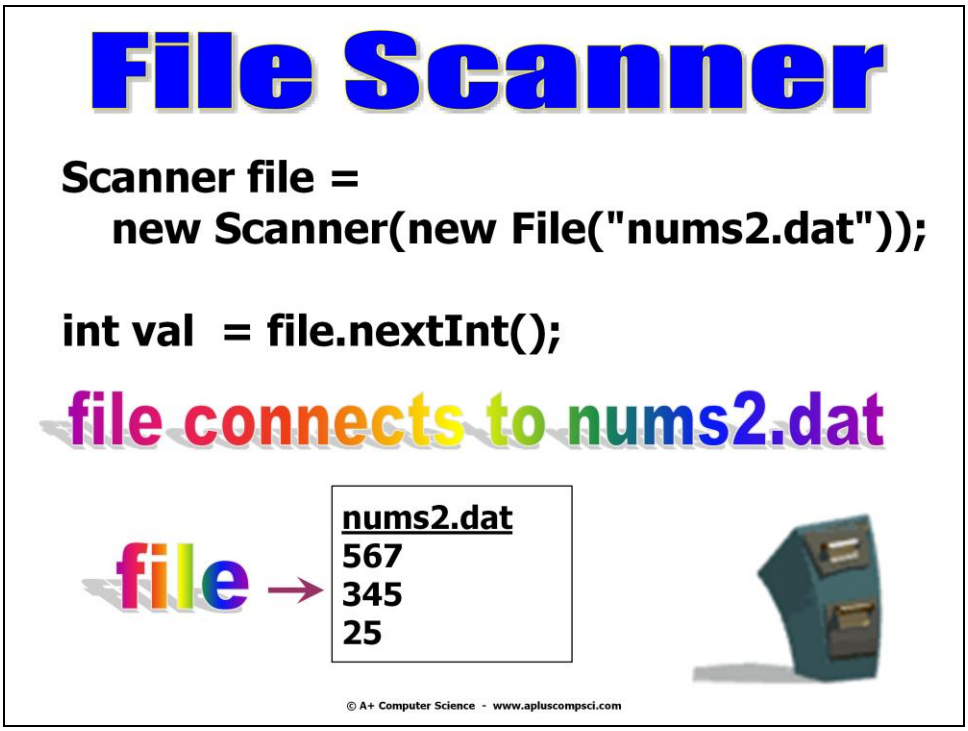


The program writes to a file.

The program reads from a file.

The file is just a container for data.

The file does not have the ability to do anything.



file stores the location/memory address of a Scanner Object that was constructed to read from the data file, `nums2.dat`.
`kb.nextInt()` calls the Scanner `nextInt()` method to read in the next integer value.

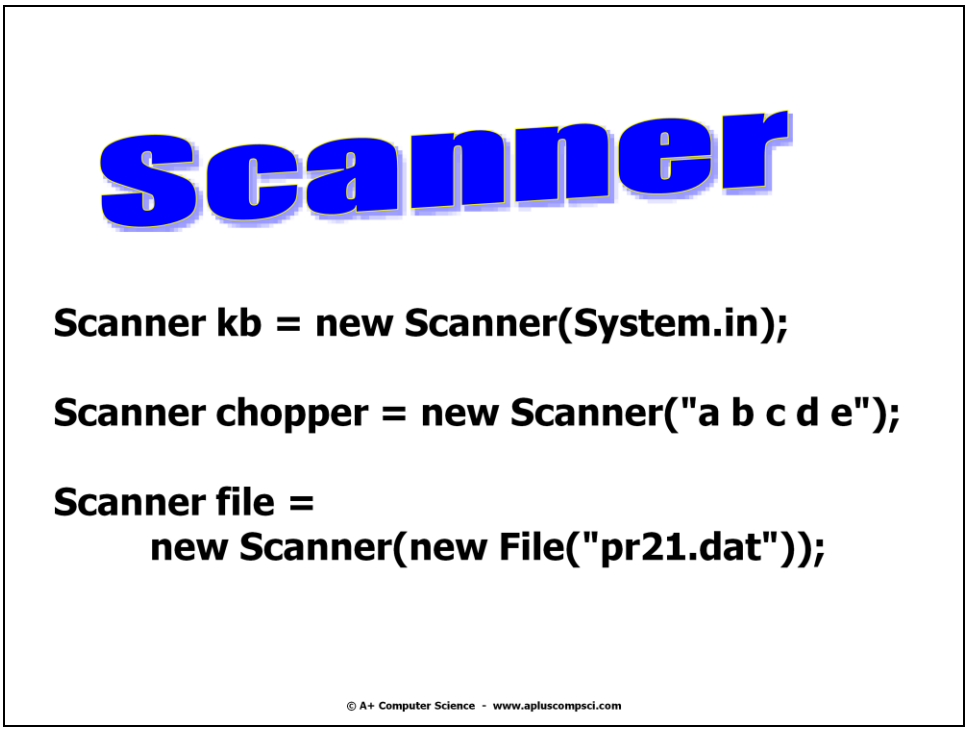
`567` would be the first integer value read in from `nums2.dat`.

Scanner

Scanner is a class in Java that can be used to read in data from the keyboard or from a data file.

Scanner can also be used to chop up Strings.

© A+ Computer Science - www.apluscompsci.com



Scanner has several constructors.

There are three constructors used most frequently.

One constructor takes in an InputStream reference.

One constructor takes in a String reference.

One constructor takes in a File parameter.

Because Scanner has multiple constructors, Scanner can be used more effectively. Scanner is very dynamic because of the different ways in which it can be used.

Scanner

```
Scanner keyboard = new Scanner(System.in);
```

object / reference — constructor — parameter

```
Scanner file
        = new Scanner(new File("prxx.dat"));
```

© A+ Computer Science - www.apluscompsci.com

Scanner has several constructors.

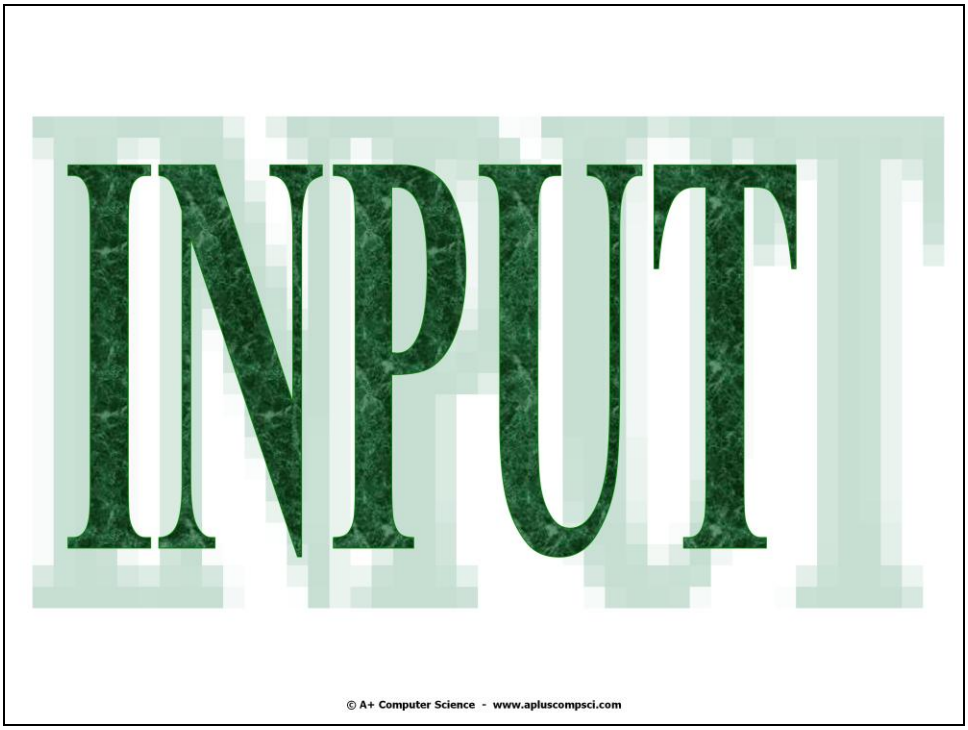
There are three constructors used most frequently.

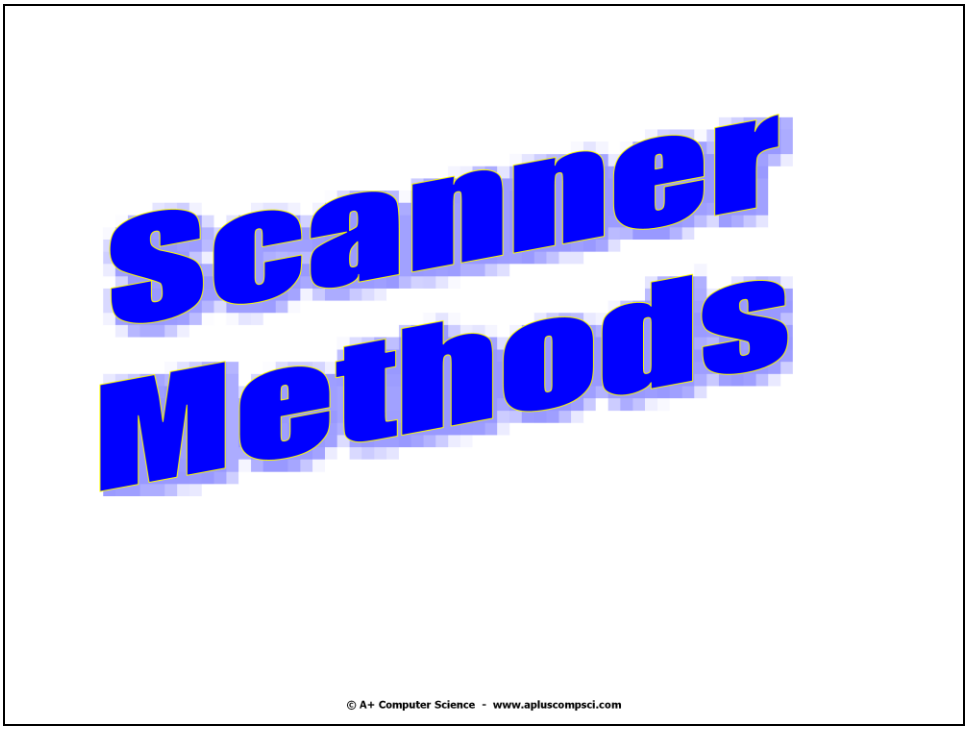
One constructor takes in an InputStream reference.

One constructor takes in a String reference.

One constructor takes in a File parameter.

Because Scanner has multiple constructors, Scanner can be used more effectively. Scanner is very dynamic because of the different ways in which it can be used.


INPUT
© A+ Computer Science - www.apluscompsci.com


Scanner
Methods
© A+ Computer Science - www.apluscompsci.com

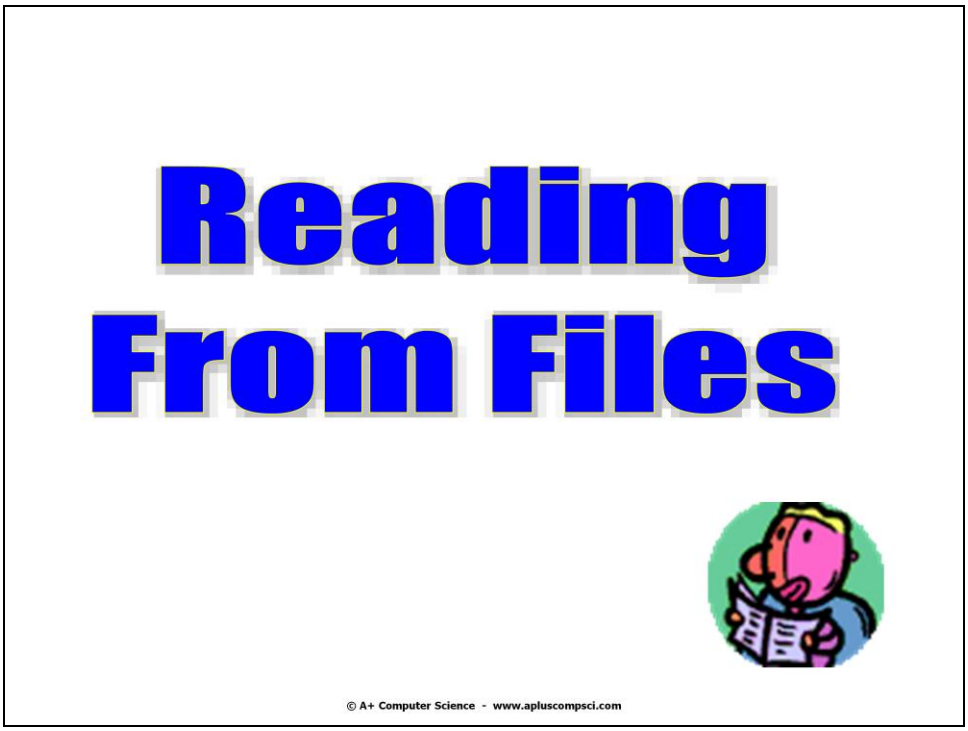
Scanner
frequently used methods

Name	Use
nextInt()	returns the next int value
nextDouble()	returns the next double value
next()	returns the next one word String
nextLine()	returns the next multi word String
hasNextInt()	checks to see if there are more ints
hasNextDouble()	checks to see if there are more doubles
hasNext()	checks to see if there are more Strings

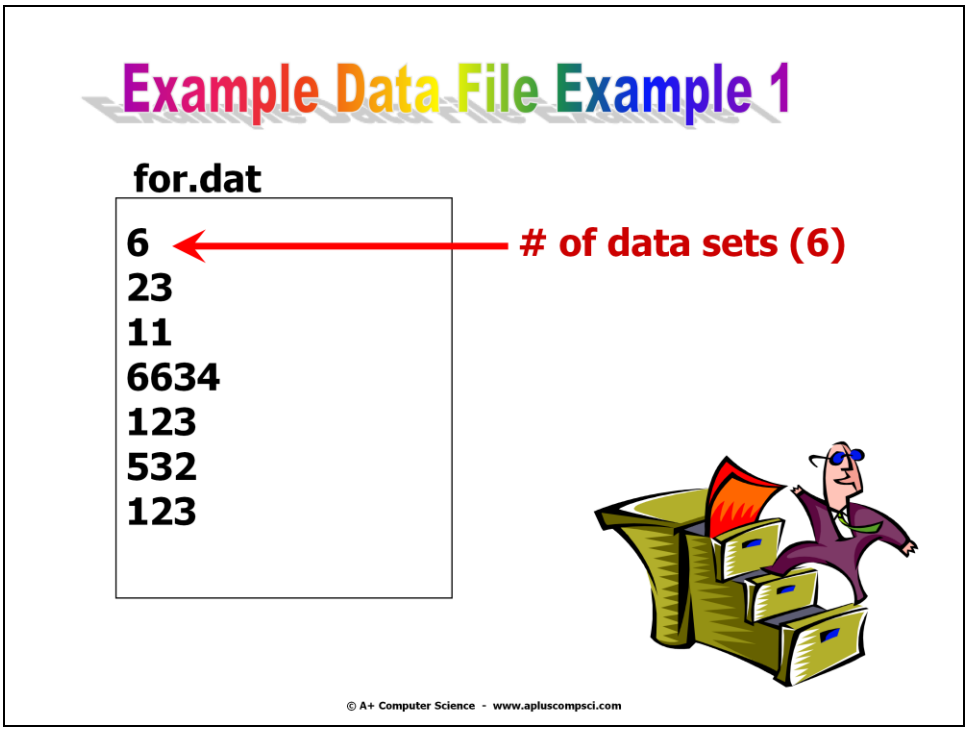
```
import  java.util.Scanner;
```

© A+ Computer Science - www.apluscompsci.com

The Scanner methods listed above include some of the most frequently used input methods and some methods that are used with loops to process multiple values from a input source.



file is a Scanner reference that stores the location/memory address of a Scanner Object constructed to read from the data file, `pr21.dat`.



Some data files contain a number of the beginning that indicates the number of values in the file.

The data file above, for.dat, has a 6 at the beginning. The 6 indicates that 6 values are present in the data file.

Reading From Files with a For Loop

```
Scanner file =
  new Scanner(new File("for.dat"));

int cnt = file.nextInt();
for(int i=0; i<cnt; i++)
{
  int num = file.nextInt();
  out.println(num);
}
```

© A+ Computer Science - www.apluscompsci.com

`cnt` will store the data value count.

The for loop will read in `cnt` values from the data file.

If there are fewer than `cnt` values in the file, an exception will be thrown.

Reading From Files with a For Loop

```
Scanner file =
  new Scanner(new File("for.dat"));

int cnt = file.nextInt();
for(int i=0; i<cnt; i++)
{
  int num = file.nextInt();
  out.println(num);
}
```

for.dat

```
3
11
5
67
```

OUTPUT

```
11
5
67
```

© A+ Computer Science - www.apluscompsci.com

`cnt` will store the data value count.

`cnt` will store the value 3.

The for loop will read in `cnt(3)` values from the data file.

Iteration 1 – 11 read in

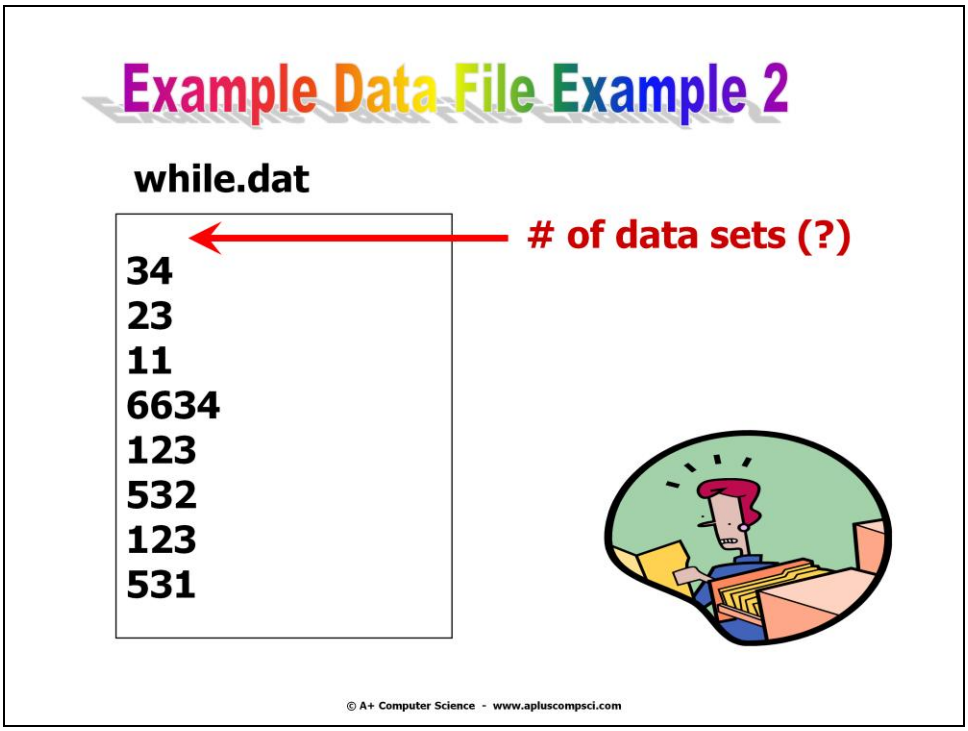
Iteration 2 – 5 read in

Iteration 3 – 67 read in

Reading From Files with a For Loop

open fileforjava

© A+ Computer Science - www.apluscompsci.com



Some data files contain values only and do not contain a value at the beginning indicating the number of values in the file.

A while loop is needed to read as long as values are present.

Reading From Files with while Loops

```
Scanner file =
  new Scanner(new File("while.dat"));

while(file.hasNextInt())
{
  int num = file.nextInt();
  out.println(num);
}
```

© A+ Computer Science - www.apluscompsci.com

Some data files contain values only and do not contain a value at the beginning indicating the number of values in the file.

A while loop is needed to read as long as values are present.

Reading From Files with while Loops

```
Scanner file =
   new Scanner(new File("while.dat"));

while(file.hasNextInt())
{
  int num = file.nextInt();
  out.println(num);
}
```

while.dat

```
11
5
67
```

OUTPUT

```
11
5
67
```

© A+ Computer Science - www.apluscompsci.com

Some data files contain values only and do not contain a value at the beginning indicating the number of values in the file.

A while loop is needed to read as long as values are present.

Iteration 1 – 11 read in

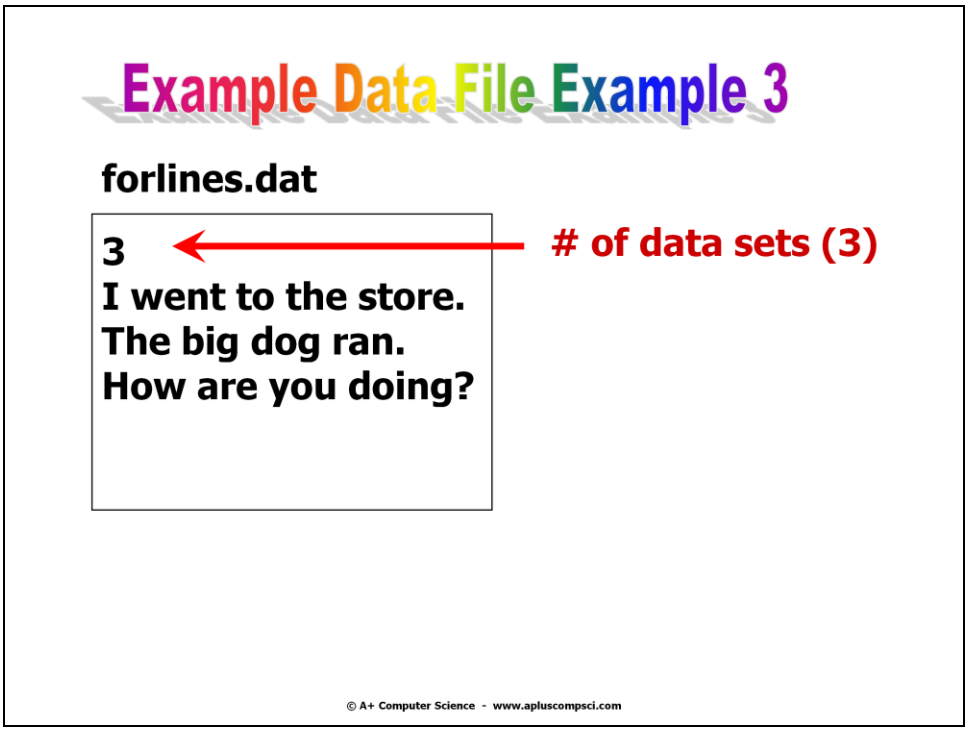
Iteration 2 – 5 read in

Iteration 3 – 67 read in

Reading From Files with while Loops

open filewhile.java

© A+ Computer Science - www.apluscompsci.com



Some data files contain a number of the beginning that indicates the number of values in the file.

The data file above, forlines.dat, has a 3 at the beginning. The 3 indicates that 3 lines of words/letters are present in the data file.

Reading Lines from a file with a for loop

```
Scanner file;
file = new Scanner(new File("forlines.dat"));
int cnt = file.nextInt();
file.nextLine();        //clear out whitespace
for(int i=0; i<cnt; i++)
{
  String sentence = file.nextLine();
  out.println(sentence);
}
```

© A+ Computer Science - www.apluscompsci.com

The example above is not much different from the previous for loop file input example.

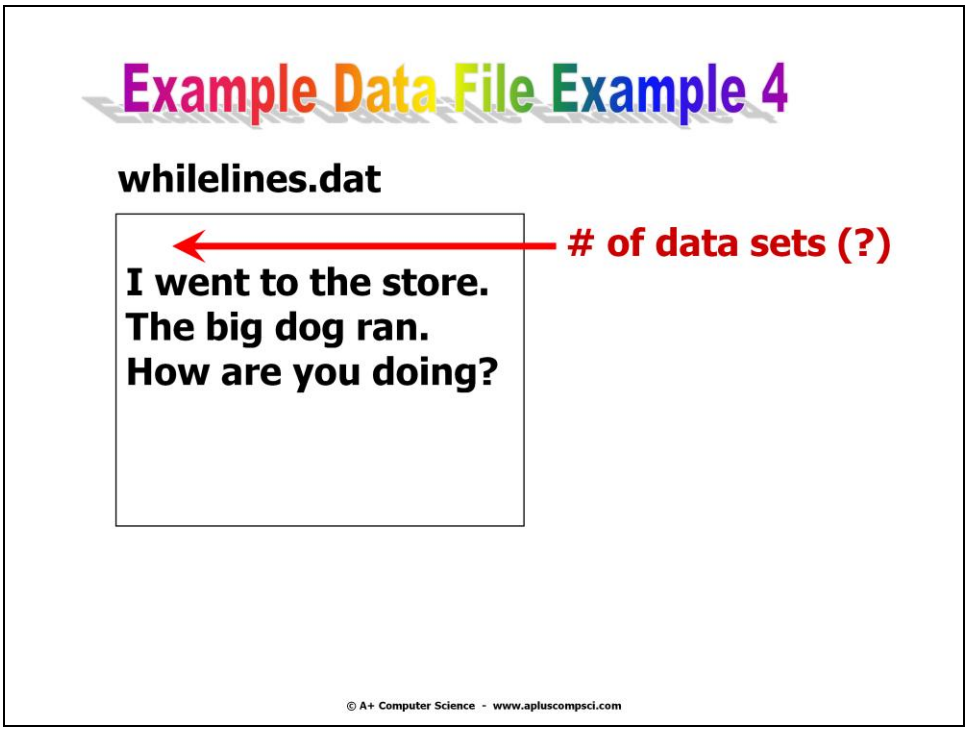
The main issue with this program is the issue of following a non-`nextLine()` method with a `nextLine()`.

The enter key left over by the `nextInt()` must be picked up.

Reading Lines from a file with a for loop

open fileforlines.java

© A+ Computer Science - www.apluscompsci.com



Some data files contain values only and do not contain a value at the beginning indicating the number of values in the file.

A while loop is needed to read as long as values are present.

Reading Lines from a file with a while loop

```
Scanner file;
file = new Scanner( new File("whilelines.dat"));
while(file.hasNext())
{
  out.println(file.nextLine());
}
```

© A+ Computer Science - www.apluscompsci.com

The while loop will contain it iterate as long as there are lines in the file.

Reading Lines from a file with a while loop

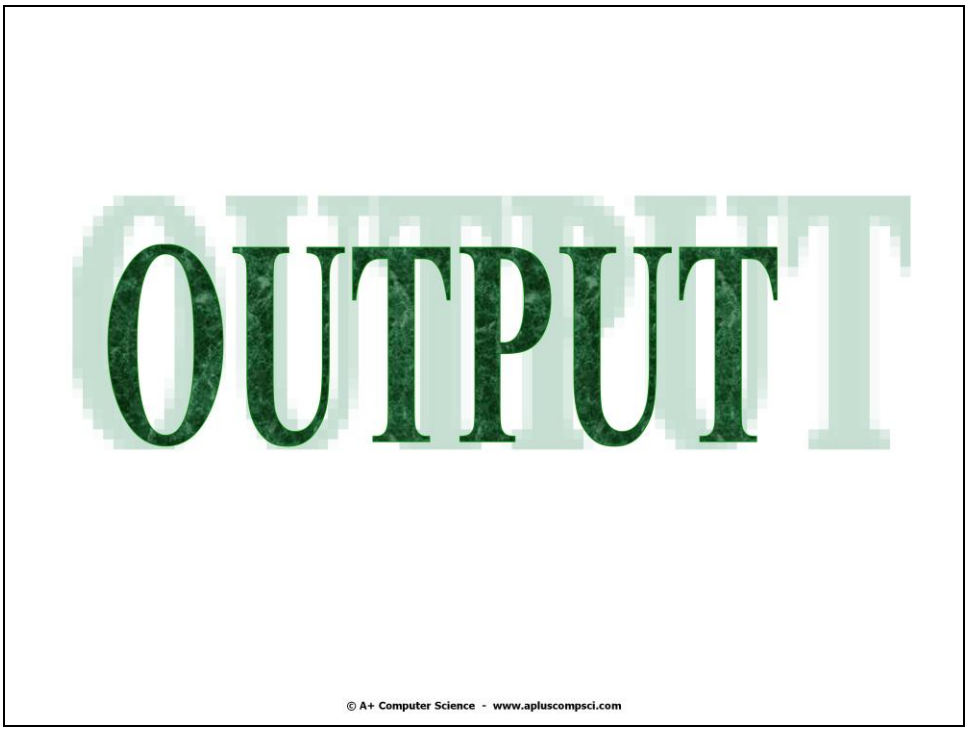
open filewhilelines.java

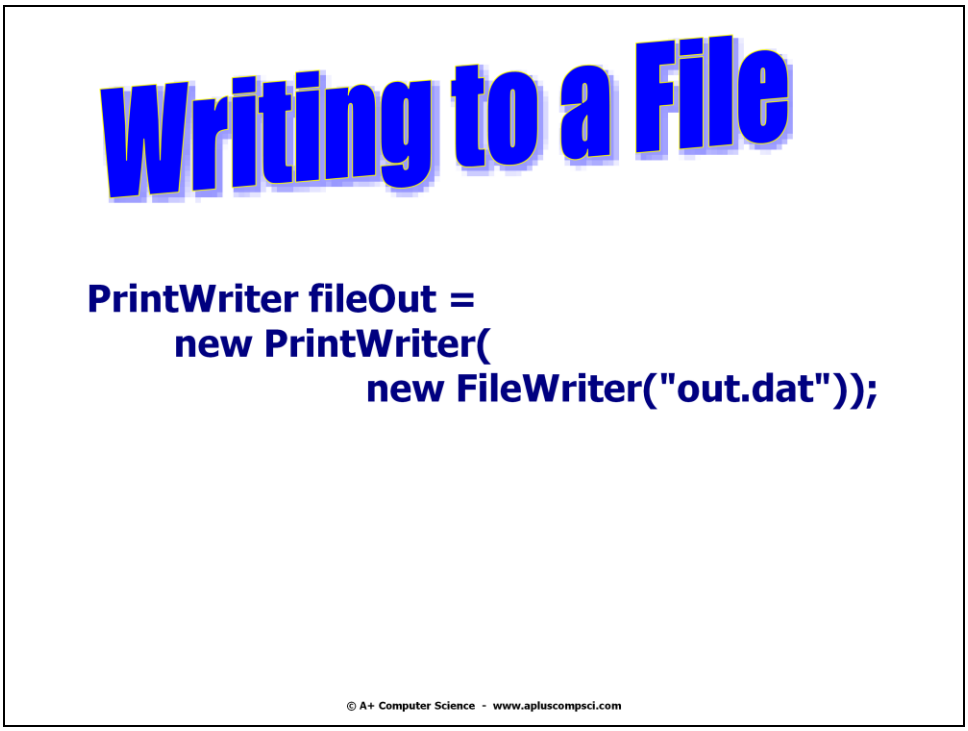
© A+ Computer Science - www.apluscompsci.com

Reading Lines from a file with a while loop and chopping them

open filewhilechop.java

© A+ Computer Science - www.apluscompsci.com


OUTPUT
© A+ Computer Science - www.apluscompsci.com



fileOut is a PrintWriter reference that is storing the location/memory address of a PrintWriter constructed to write to out.dat.

loops for output

```
PrintWriter fileOut =
        new PrintWriter(
                new FileWriter("out.dat"));

for(int i=0; i<10; i++)
{
   fileOut.println(i);
}
fileOut.close();
```

© A+ Computer Science - www.apluscompsci.com

out.dat will contain the numbers `0,1,2,3,4,5,6,7,8,9` once the loop terminates.

The file must be closed once the writing process is over.

Closing the FILE

Every file you open you must also close.

file.close();

kb.close();

© A+ Computer Science - www.apluscompsci.com

open fileout.java

© A+ Computer Science - www.apluscompsci.com


Start work
on the labs
© A+ Computer Science - www.apluscompsci.com