

CS 207 - Module 1
Example: Set-up to read data from
a text file

Example: Set-up for 'read data from a text file'

- 1) set import statements
 - a. three classes needed
 - i. class Scanner, to access the data;
 - ii. class File, to make a connection to the physical data file
 - iii. class FileNotFoundException, which is required by the File class whenever code is to be written to access a file
 1. note: ii and iii can be written with a wildcard character, as both are in the java.io package.
- 2) create the class and method main
- 3) add try-catch block
- 4) complete catch (note: code will not compile at this point)
- 5) create Scanner object to access text file (note: code will now compile)
- 6) add while loop to access each record in text file
- 7) create a String to store a record
- 8) perform program logic upon read-in string
 - a. in this example, display the currently read-in string

End of Example "Set-up for 'read data from file' ".

See next pages for each step shown with corresponding code.

Set-up to read from a data file

- 1) set import statements

```
import java.util.Scanner;  
import java.io.*;
```

- 2) create the class and method main

```
import java.util.Scanner;  
import java.io.*;  
  
public class ReadDataFromFileSetUp  
{  
    public static void main(String [] args)  
    {  
    }  
}
```

- 3) add try-catch block

```
import java.util.Scanner;  
import java.io.*;  
  
public class ReadDataFromFileSetUp  
{  
    public static void main(String [] args)  
    {  
        try  
        {  
        }  
        catch( )  
        {  
        }  
    }  
}
```

4) complete catch (note: code will not compile at this point)

```
import java.util.Scanner;
import java.io.*;

public class ReadDataFromFileSetUp
{
    public static void main(String [] args)
    {
        try
        {

        }
        catch(FileNotFoundException fnf)
        {
            System.out.println("no file" + fnf);
        }
    }
}
```

5) create Scanner object to access text file (note: code will now compile)

```
import java.util.Scanner;
import java.io.*;

public class ReadDataFromFileSetUp
{
    public static void main(String [] args)
    {
        try
        {
            Scanner fScanner = new Scanner(new File("countries.txt"));
        }
        catch(FileNotFoundException fnf)
        {
            System.out.println("no file" + fnf);
        }
    }
}
```

6) add while loop to access each record in text file

```
import java.util.Scanner;
import java.io.*;

public class ReadDataFromFileSetUp
{
    public static void main(String [] args)
    {
        try
        {
            Scanner fScanner = new Scanner(new File("countries.txt"));

            while(fScanner.hasNextLine())
            {

            }
        }
        catch(FileNotFoundException fnf)
        {
            System.out.println("no file" + fnf);
        }
    }
}
```

7) create a String to store a record

```
import java.util.Scanner;
import java.io.*;

public class ReadDataFromFileSetUp
{
    public static void main(String [] args)
    {
        try
        {
            Scanner fScanner = new Scanner(new File("countries.txt"));

            while(fScanner.hasNextLine())
            {
                String line = fScanner.nextLine();
            }
        }
        catch(FileNotFoundException fnf)
        {
            System.out.println("no file" + fnf);
        }
    }
}
```

8) perform program logic upon read-in string

```
import java.util.Scanner;
import java.io.*;

public class ReadDataFromFileSetUp
{
    public static void main(String [] args)
    {
        try
        {
            Scanner fScanner = new Scanner(new File("countries.txt"));

            while(fScanner.hasNextLine())
            {
                String line = fScanner.nextLine();

                System.out.println(line);
            }
        }
        catch(FileNotFoundException fnf)
        {
            System.out.println("no file" + fnf);
        }
    }
}
```

Module 1

Practice Exercises

Exercise 1

Using countries.txt, display each record on a separate line, with each data item separated by a space.

Program output:

```
USA 203 Washington
Albania 132 Tirana
Afghanistan 111 Kabul
Barbados 97 Bridgetown
Bahrain 123 Manama
```

Exercise 2

Using integers.txt, display the sum of all values in the text file.

Exercise 3

Using integers.txt, display all integers, each on a separate line

Exercise 4

Using sentences.txt, display each sentence in the file, such that each sentence appears on a separate line, and is separated from the previous sentence by an empty line space.