# Creating a text file

- When programming, we may be required to read and write out data to files.
- In Java, when reading and writing, the input and output files are saved as text files.
- We will learn how to write and read Strings from text files.

## Sequential Files

- **Data** that is stored permanently is saved in sequential (text files) or binary files.
- This data exists even while a program is not running.
  - Examples:

## For Example:

- The following data file contains the marks of a student (file name: marks.txt):

```
72.4
65.2
80.3
```

## Reading from a text file

## Reading from a text file

To create a stream that reads from a file…

1) Construct an object of the FileReader class (from package java.io) which is an object which contains the path and name of the file that is to be read from.
2) Wrap the stream in a BufferedReader class that buffers the characters from the FileReader
3) Read from a file using the readLine method of the BufferedReader class
   – If the end of the file is reached, readLine returns the value **null**

# Example 1: Syntax

```
import java.io.*;  // FileReader is found in this class

class Test {
        public static void main (String[] args) throws IOException //discovers errors
      when.. ..reading text files
                {
                String x;   // Create a varaible to store data
                FileReader fr=new FileReader("g:\\marks.txt");  //1) FileReader
                BufferedReader br = new BufferedReader(fr);   //2) BufferedReader

                x = br.readLine();    //3) readLine returns a line of data as a string
                System.out.println(x);
                x = br.readLine();    //readLine returns the next line of data as a string
                System.out.println(x);
                x = br.readLine();    //readLine returns the next line of data as a string
                System.out.println(x);

                br.close();    // close br (close the file)
}
```

# What would be printed here?

```
import java.io.*;  // FileReader is found in this class

class Test {
        public static void main (String[] args) throws IOException {
                String x;   // Create a varaible to store data
                FileReader fr=new FileReader("g:\\marks.txt");  // 1 ) FileReader
                BufferedReader br = new BufferedReader(fr);   // 2) BufferedReader

                x = br.readLine();    //3) readLine returns a line of data as a string
                System.out.println(x);
                x = br.readLine();    //readLine returns the next line of data as a string
                System.out.println(x);
                x = br.readLine();    //readLine returns the next line of data as a string
                System.out.println(x);
                x = br.readLine();    //readLine returns the next  line of data as a string
                System.out.println(x);

                br.close();  //close method and flushes data buffer and closes connection to file
….
```

# Example 2: Using  a loop

```
//This segment of code will continue to read in files until the end of file is reached

import java.io.*;
public class Test {


      public static void main(String[] args) throws IOException {
            String x;
            **try{**
                  FileReader fr=new FileReader("g:\\marks.txt");
                  BufferedReader br = new BufferedReader(fr);
                  while ((x = br.readLine()) != null) {                                    //reads data
    into x and then compares x  to null
                        System.out.println(x);
                  }
                  br.close();

            }**catch (IOException e){}     //**
            **catch (NumberFormatException e){}   //catches formatting errors in the text and
    ignores them**


      }
}
```

# Note:

- One of following must be present when reading from text files
  **1) throws IOException**
  **2) catch (IOException e){}     //**
  **catch (NumberFormatException e){}**

1) Will show errors when reading text files
2) will catch errors when reading text files

## Writing to a text file

## Writing to a text file

To write to a text file…

1) Create a FileWriter object that writes characters to an output stream to a file. If the file does not exist, the constructor creates it
2) Create a PrintWriter object which contains println and print methods to write to a file.
3) Use println/print to write to a file

# Example: Syntax

```
import java.io.*;  // FileWriter is found in this class

public class Test {

        public static void main(String[] args) throws IOException {
                FileWriter fw = new FileWriter("g:\\Sample.txt");    // 1) FileWriter
                PrintWriter pw = new PrintWriter (fw);     // 2) PrintWriter

                pw.println("231231");    // 3) Write to file
                pw.println("world2");
                pw.close();
        }
}
```

# Sample.txt

231231
world2

## Useful String Commands

https://www.w3schools.com/java/java_ref_string.asp

Useful methods to use:

charAt()
substring()