**CSV Files**

* The CSV stands for Comma-Separated Values.
* It is a simple file format which is used to store tabular data in simple text form, like: spreadsheet or database.
* We can create CSV files in excel or notepad.
* We can export or import this file from M.S office or excel.
* Here the data is stored in column by column, and split by a separator i.e “ , “ (comma).
* Example: [Amar, 42, CSE, pass, 8.6]

**Handling CSV Files in Java**

* First step is to create a CSV File using: 1) Excel 2) Notepad.
* And then select the **save as type** menu CSV (Common delimited) and then click **save** button.
* In notepad you have save the file with the **.csv** extension.

**Reading CSV Files in Java**

* There are three ways to read CSV file:
  + **Scanner** class
  + **String.split()**
  + Using **OpenCSV** API

Java Scanner class:

* It provides different methods by which we can read CSV files in java.
* Scanner class provides a constructor that produces values scanned from specific file. It breaks the input into tokens whenever it finds whitespace.
* Ex: “Hello world” it splits into two tokens: “Hello”,

” world”.

* We can read each token using some next() methods like:
  + next()
  + nextInt()
  + nextDouble()
* It uses a delimiter pattern based on spaces, tabs, newlines. Use useDelimiter() to specify some different delimiter i.e comma for CSV files.
* Example :

package fileHandling;

import java.io.\*;

import java.util.Scanner;

public class UsingScannerClass {

public static void main(String[] args) {

try

{

Scanner object = new Scanner(new File("D:\\CSV file.csv"));

object.useDelimiter(" , ");

while(object.hasNext())

{

System.out.print(object.next());

}

object.close();

}

catch(Exception e)

{

System.out.print("Catch block");

e.getCause();

}

}

}

Split() Method:

* It is used to split a string into an array of substrings or a line into array of string based on some delimiter.
* It uses some delimiter (any character) to split the line.
* Syntax:

String[] result = splitStringObject.split(delimiter);

* Example: {“joe, doe, developer”} this is a line. Here the method identifies each comma in the line and uses it as boundary to create a new element in a array. i.e {“joe” , “doe” , “developer”}
* Example:

package fileHandling;

//BufferedReader and FileReader used in reading files

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

// this class is demonstrating how to read a CSV File

public class BufferReaderClass {

public static void main(String[] args) {

//line variable holds each line, read from the csv file

String line="";

//splitBy here is a delimiter used to split each line

String splitBy=",";

//exception block

try

{

//BufferReader reads and provides buffering.FileReader is specified with file path and is used to read the file

BufferedReader bufferobject= new BufferedReader(new FileReader("D:\\CSV file.csv"));

//while loop reads each line from the file with the help of readLine() until the end of the file

while((line=bufferobject.readLine())!=null)

{

//here split() method is used

String[] employee = line.split(splitBy);

if(employee.length==6)

{

System.***out***.println("Employee [First Name= "+employee[0]+", Last Name= "+employee[1]+", Designation= "+employee[2]+", Contact= "+employee[3]+", Salary= "+employee[4]+", City= "+employee[5]+"]");

}

else

{

System.***out***.println("Incomplete line");

}

}

}

catch(Exception e)

{

System.***out***.println("Catch block ");

e.getCause();

e.printStackTrace();}

}

}

Using OpenCSV API:

* It is third party app