**What is a CSV File?**

A csv file is a kind of flat file used to store the data. In this each row contains data separated by comma.  
For example,

|  |  |
| --- | --- |
| 1  2  3  4 | 20,hi,99  3,4,5  a,b,c  3,4,1 |

**Creating a Class CSVReader to read Data from CSV File**

Let’s create a class CSVReader that provides API to read data from a CSV File

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | /\*  \* A class to read data from a csv file.  \*/  class CSVReader  {  std::string fileName;  std::string delimeter;    public:  CSVReader(std::string filename, std::string delm = ",") :  fileName(filename), delimeter(delm)  { }    // Function to fetch data from a CSV File  std::vector<std::vector<std::string> > getData();  }; |

It accepts the filename as constructor argument and provides a member function that reads the content of CSV file and returns the data in a vector of vector<std::string> i.e.

|  |  |
| --- | --- |
| 1  2 | // Function to fetch data from a CSV File  std::vector<std::vector<std::string> > getData(); |

All comma separated elements in a single row will be stored in vector of strings i.e.

|  |  |
| --- | --- |
| 1 | std::vector<std::string> |

Now Data of each row i.e. vector<string> will be stored in another vector i.e.

|  |  |
| --- | --- |
| 1 | std::vector<std::vector<std::string> > |

This member function will parse the content of csv file line by line and store data in vector< vector <string> > i.e.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | /\*  \* Parses through csv file line by line and returns the data  \* in vector of vector of strings.  \*/  std::vector<std::vector<std::string> > CSVReader::getData()  {  std::ifstream file(fileName);    std::vector<std::vector<std::string> > dataList; // datalist is vector of vector of string    std::string line = "";  // Iterate through each line and split the content using delimeter  while (getline(file, line))  {  std::vector<std::string> vec; // vec is vector of string  boost::algorithm::split(vec, line, boost::is\_any\_of(delimeter));  dataList.push\_back(vec);  }  // Close the File  file.close();    return dataList;  } |

Let’s create an object of class CSVReader class and read the data from csv file i.e.

|  |  |
| --- | --- |
| 1  2  3  4  5 | // Creating an object of CSVWriter  CSVReader reader("example.csv");    // Get the data from CSV File  std::vector<std::vector<std::string> > dataList = reader.getData(); |

Complete example is as follows,

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68 | #include <iostream>  #include <fstream>  #include <vector>  #include <iterator>  #include <string>  #include <algorithm>  #include <boost/algorithm/string.hpp>    /\*  \* A class to read data from a csv file.  \*/  class CSVReader  {  std::string fileName;  std::string delimeter;    public:  CSVReader(std::string filename, std::string delm = ",") :  fileName(filename), delimeter(delm)  { }    // Function to fetch data from a CSV File  std::vector<std::vector<std::string> > getData();  };    /\*  \* Parses through csv file line by line and returns the data  \* in vector of vector of strings.  \*/  std::vector<std::vector<std::string> > CSVReader::getData()  {  std::ifstream file(fileName);    std::vector<std::vector<std::string> > dataList;    std::string line = "";  // Iterate through each line and split the content using delimeter  while (getline(file, line))  {  std::vector<std::string> vec;  boost::algorithm::split(vec, line, boost::is\_any\_of(delimeter));  dataList.push\_back(vec);  }  // Close the File  file.close();    return dataList;  }  int main()  {  // Creating an object of CSVWriter  CSVReader reader("example.csv");    // Get the data from CSV File  std::vector<std::vector<std::string> > dataList = reader.getData();    // Print the content of row by row on screen  for(std::vector<std::string> vec : dataList)  {  for(std::string data : vec)  {  std::cout<<data << " , ";  }  std::cout<<std::endl;  }  return 0;    } |

**Output:**

|  |  |
| --- | --- |
| 1  2  3  4 | 20 , hi , 99 ,  3 , 4 , 5 ,  a , b , c ,  3 , 4 , 1 , |

To Compile the above code use following command in Linux,

**g++ –std=c++11 example.cpp**