# Department of Computing

# School of Electrical Engineering and Computer Science

**CS-250: Data Structure and Algorithms**

**Class: BSCS 10C**

**Lab 06:  Reading CSV Files**

**Date: 29th October, 2021**

**Time: 9:00 am – 11:50 am**

# Instructor: Prof. Dr. Faisal Shafait

# Lab Engineer: Mr. Aftab Farooq

# Lab 06: CSV Reader

**Introduction**

CSV Reader is used to view CSV files, it can quickly open and format the data in the (\*.csv) files and help you to easily browse and view data stored in CSV files.

**Objectives**

The purpose of this lab is to read CSV files and implement certain operations on the data

**Description**

**CSV** is a simple **file** format used to store tabular data, such as a spreadsheet or database. **Files** in the **CSV** format can be imported to and exported from programs that store data in tables, such as Microsoft Excel or OpenOffice Calc. CSV stands for "comma-separated values".

**Tools/Software Requirement**

Visual Studio C++

**Lab Tasks**

**Task A:**

Consider the regional schools record file: Tehsil Schools.xlsx – Read that file in your program using CSVRow class (see hint) and load all the data in a vector of strings. The filename should be specified at the command line and not hardcoded in the program.

|  |
| --- |
| Code |
| #include <iostream>  #include <string>  #include <vector>  #include <sstream>  #include <fstream>  using **namespace** std;  **class** CSVRow*//class for making vector and reading rows*  {  **public:**  vector<string> m\_data;*//vector for storing the whole CSV*  **int** getsize() **const**  {  return m\_data.size();*//get size of vector*  }  **void** ReadNextRow(ifstream **&**str)*//taking address of the string as an argument*  {  string line;  getline(str, line);*//strings uptil comma will combine to make a line*  stringstream lineStream(line);  string cell;*//cell will be uptil comma*  while (getline(lineStream, cell, ','))  {  m\_data.push\_back(cell);*//storing cells*  m\_data.push\_back(" ");  }  m\_data.push\_back("\n");  }  **void** display()*//displayer function for the whole vector*  {  for (**int** i = 0; i < m\_data.size();i++)  {  cout << m\_data[i];  }  }  };  **int** main()  {  ifstream myfile; *//used to open and read file*  string fileName; *//takes in path of the file on command line*  cout << "Enter File Name: ";  getline(cin, fileName);  myfile.open(fileName, ios::in); *//opening the csv file*  CSVRow data;  while(myfile.peek()!=EOF) *//loop until the end of the file*  {  data.ReadNextRow(myfile); *//reading each row uptil the end of the file*  }  data.display(); *//using the displayer function*  getchar();  return 0;  } |
| Output |
| Text  Description automatically generated |

**Task B:**

Print the names of all schools for which the number of students passed in 10th Class Exam is ZERO.

|  |
| --- |
| Code |
| #include <iostream>  #include <string>  #include <vector>  #include <sstream>  #include <fstream>  using **namespace** std;  **class** CSVRow *//class for making vector and reading rows*  {  **public:**  vector<string> m\_data; *//vector for storing the whole CSV*  **int** getsize() **const**  {  return m\_data.size(); *//get size of vector*  }  **void** ReadNextRow(ifstream **&**str) *//taking address of the string as an argument*  {  string line;  getline(str, line); *//strings uptil \n character will make a line*  stringstream lineStream(line);  string row; *//cell will be uptil comma*  while (getline(lineStream, row, '\n'))  {  m\_data.push\_back(row); *//storing the vector row wise*  }  m\_data.push\_back("\n");  }  **void** display() *//displayer function for the whole vector*  {  vector<string> cell;  string temp; *//temp string to store cells*  for (**int** i = 0; i < m\_data.size(); i++)  {  stringstream line(m\_data[i]);  while (getline(line, temp, ','))  {  cell.push\_back(temp); *//cell vector storing data cell wise*  }  if (cell.size() == 13 || cell.size() == 14) *//the rows which have 13 or 14 cells are the ones which*  {  if (stoi(cell[8]) == 0)*//where no of students passed is zero*  { *//cell 8 is the number of students passed in 10th*  cout << cell[1] << endl; *//cell[1] contains names of schools*  }  }  cell.clear(); *//clear the cell to save the memory*  }  }  };  **int** main()  {  ifstream myfile; *//used to open and read file*  string path; *//takes in path of the file on command line*  cout << "Enter the path of the CSV file to be read: ";  getline(cin, path);  myfile.open(path, ios::in); *//opening the csv file*  CSVRow data;  while (!myfile.eof()) *//loop until the end of the file*  {  data.ReadNextRow(myfile); *//reading each row uptil the end of the file*  }  cout << "The names of schools schools for which the number of students passed in 10th Class Exam is ZERO are : " << endl;  data.display();  getchar();  return 0;  } |
| Output |
| **Text  Description automatically generated** |

**Task C:**

Find the percentage of large sized schools w.r.t. student enrolment, assuming a school to be large if it had more than 50 Students in 9th (2012) as per Registration.

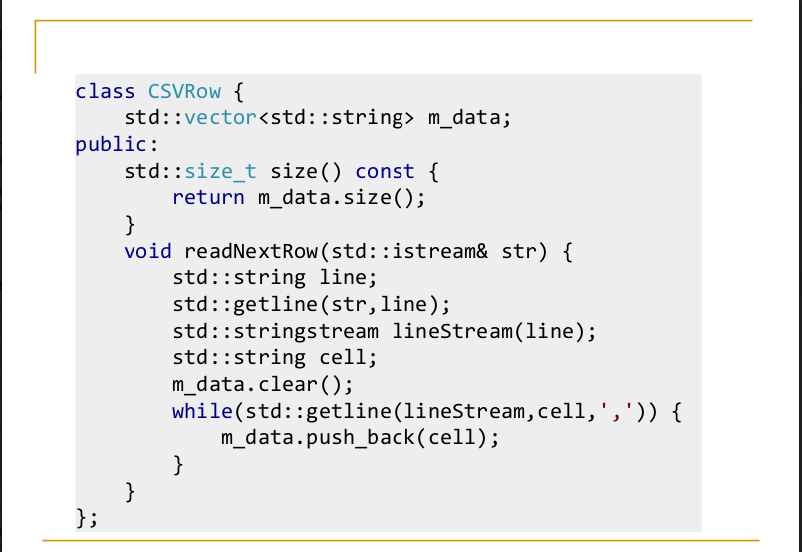
|  |
| --- |
| Code |
| #include <iostream>  #include <string>  #include <vector>  #include <sstream>  #include <fstream>  using **namespace** std;  **class** CSVRow *//class for making vector and reading rows*  {  **public:**  vector<string> m\_data; *//vector for storing the whole CSV*  **int** getsize() **const**  {  return m\_data.size(); *//get size of vector*  }  **void** ReadNextRow(ifstream **&**str) *//taking address of the string as an argument*  {  string line;  getline(str, line); *//strings uptil \n character will make a line*  stringstream lineStream(line);  string row; *//cell will be uptil comma*  while (getline(lineStream, row, '\n'))  {  m\_data.push\_back(row); *//storing the vector row wise*  }  m\_data.push\_back("\n");  }  **void** display() *//displayer function for the whole vector*  {  vector<string> cell;  string temp; *//temp string to store cells*  for (**int** i = 0; i < m\_data.size(); i++)  {  stringstream line(m\_data[i]);  while (getline(line, temp, ','))  {  cell.push\_back(temp); *//cell vector storing data cell wise*  }  if (cell.size() == 13 || cell.size() == 14) *//the rows which have 13 or 14 cells are the ones which*  {  if (stoi(cell[6]) > 50) *//large schools have No. of Students in 9th (2012) as per Registeration greater than 50*  {  cout << cell[1] << " " << cell[9] << "%" << endl; *//cell[9] has all percentages in it*  }  }  cell.clear(); *//clear the cell to save the memory*  }  }  };  **int** main()  {  ifstream myfile; *//used to open and read file*  string path; *//takes in path of the file on command line*  cout << "Enter the path of the CSV file to be read: ";  getline(cin, path);  myfile.open(path, ios::in); *//opening the csv file*  CSVRow data;  while (!myfile.eof()) *//loop until the end of the file*  {  data.ReadNextRow(myfile); *//reading each row uptil the end of the file*  }  cout << "The percentage of large sized schools w.r.t. is given below: " << endl;  data.display();  getchar();  return 0;  } |
| Output |
| Text  Description automatically generated |

**Task D:**

Among all large sized schools, print the name of the school with the highest % dropout.

|  |
| --- |
| Code |
| #include <iostream>  #include <string>  #include <vector>  #include <sstream>  #include <fstream>  using namespace std;  class CSVRow *//class for making vector and reading rows*  {  public:      vector<string> m\_data; *//vector for storing the whole CSV*      int getsize() const      {          return m\_data.size(); *//get size of vector*      }      void ReadNextRow(ifstream &str) *//taking address of the string as an argument*      {          string line;          getline(str, line); *//strings uptil \n character will make a line*          stringstream lineStream(line);          string row; *//cell will be uptil comma*          while (getline(lineStream, row, '\n'))          {              m\_data.push\_back(row); *//storing the vector row wise*          }          m\_data.push\_back("\n");      }      void display() *//displayer function for the whole vector*      {          vector<string> cell;          string temp; *//temp string to store cells*          float max = 0;          float temporary = 0;          for (int i = 0; i < m\_data.size(); i++)          {              stringstream line(m\_data[i]);              while (getline(line, temp, ','))              {                  cell.push\_back(temp); *//cell vector storing data cell wise*              }              if (cell.size() == 13 || cell.size() == 14) *//the rows which have 13 or 14 cells are the ones which*              {                  if (stoi(cell[6]) > 50)*//large sized schools logic*                  {  *//finding maximum from the column of %dropouts*                      temporary = stof(cell[12]);                      if (temporary >= max)                      {                          max = temporary;                      }                  }              }              cell.clear(); *//clear the cell to save the memory*          }  *//traversing again*          for (int i = 0; i < m\_data.size(); i++)          {              stringstream line(m\_data[i]);              while (getline(line, temp, ','))              {                  cell.push\_back(temp); *//cell vector storing data cell wise*              }              if (cell.size() == 13 || cell.size() == 14) *//the rows which have 13 or 14 cells are the ones which*              {  *//the percentage with the maximum dropout*                  if (stof(cell[12]) == max)                  {                      cout << cell[1] << "     " << cell[12] <<"%"<< endl;*//print it*                  }              }              cell.clear(); *//clear the cell to save the memory*          }      }  };  int main()  {      ifstream myfile; *//used to open and read file*      string path; *//takes in path of the file on command line*      cout << "Enter the path of the CSV file to be read: ";      getline(cin, path);      myfile.open(path, ios::in); *//opening the csv file*      CSVRow data;      while (!myfile.eof()) *//loop until the end of the file*      {          data.ReadNextRow(myfile); *//reading each row uptil the end of the file*      }      cout << "School:             "           << "%Dropout:  " << endl;      data.display();      getchar();      return 0;  } |
| Output |
|  |

**Hint :**



**Lab Grading:**

|  |  |
| --- | --- |
| **Task** | **Marks** |
| Lab Viva/Quiz | 5 |
| Comments/ Indentation | 2 |
| Solution Document | 2 |
| Output Screen Shots | 1 |
| -- | -- |
| Total | 10 |

**Deliverables**

Compile a single word document by filling in the solution part and submit this Word file on LMS. The name of word document should follow this format. i.e. **YourFullName(reg)\_Lab#.** This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva related to the tasks. The viva has a weightage of 5 marks. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems discuss it by emailing it to [aftab.farooq@seecs.edu.pk](mailto:aftab.farooq@seecs.edu.pk).

**Note:** Students are required to upload the lab on LMS before deadline.

Use proper indentation and comments. Lack of comments and indentation will result in deduction of marks.