# 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**

**Muhammad Ahmed Fraz**

**332779**

**BSCS-!0C**

# 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<vector>  #include<string>  #include<sstream>  #include<fstream>  using namespace std;  class CSVRow {  public:  vector<string> data;  size\_t size() const {  return data.size();  }  void readNextRow(istream& str) {  string line;  getline(str, line);  stringstream lineStream(line);  string cell;  data.clear();  while (getline(lineStream, cell, ','))  {  data.push\_back(cell);  }  }  void displayFile(string path) {  ifstream myFile;  myFile.open(path);  while (myFile) {  readNextRow(myFile);  for (int i = 0; i < (int)size(); i++) {  cout << data[i];  }  cout << endl;  }    }  };  int main() {  cout << "Enter the name of the file:" << endl;  string fileName;  getline(cin, fileName);  ifstream File;  CSVRow\* csv = new CSVRow();  File.open(fileName);  csv->readNextRow(File);  csv->displayFile(fileName);  } |

**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.

|  |
| --- |
| #include<iostream>  #include<vector>  #include<string>  #include<sstream>  #include<fstream>  using namespace std;  class CSVRow {  public:  vector<string> data;  size\_t size() const {  return data.size();  }  void readNextRow(istream& str) {  string line;  getline(str, line);  stringstream lineStream(line);  string cell;  data.clear();    while (getline(lineStream, cell, ','))  {  data.push\_back(cell);    }    }  void displayFile(string path) {  ifstream myFile;  myFile.open(path);  while (!myFile.eof()) {  readNextRow(myFile);  for (int i = 0; i < (int)size(); i++) {  cout << data[i];  }    cout << endl;  }      }  bool isDigit(const string& s)  { string::const\_iterator iterator= s.begin();  while (iterator != s.end() && isdigit(\*iterator))  ++iterator;  return !s.empty() && iterator == s.end();  }  void displaySchools(string path) {  ifstream myFile;  vector<string> temp1;  vector<string> temp2;  int temp;  myFile.open(path);  while (!myFile.eof()) {  readNextRow(myFile);    if (isDigit((data[8])) ){  temp = stoi(data[8]);  if (temp == 0) {  cout << "School Name: " << data[1] << endl;  cout<< "No. of students: " << temp << endl;  cout << endl;  }  }          }  }  };  int main() {  cout << "Enter the name of the file:" << endl;  string fileName;  getline(cin, fileName);  ifstream File;  CSVRow\* csv = new CSVRow();  File.open(fileName);  csv->readNextRow(File);  csv->displaySchools(fileName);  } |

**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.

|  |
| --- |
| #include<iostream>  #include<vector>  #include<string>  #include<sstream>  #include<fstream>  using namespace std;  class CSVRow {  public:  vector<string> data;  size\_t size() const {  return data.size();  }  void readNextRow(istream& str) {  string line;  getline(str, line);  stringstream lineStream(line);  string cell;  data.clear();    while (getline(lineStream, cell, ','))  {  data.push\_back(cell);    }    }  void displayFile(string path) {  ifstream myFile;  myFile.open(path);  while (!myFile.eof()) {  readNextRow(myFile);  for (int i = 0; i < (int)size(); i++) {  cout << data[i];  }    cout << endl;  }      }  bool isDigit(const string& s)  { string::const\_iterator iterator= s.begin();  while (iterator != s.end() && isdigit(\*iterator))  ++iterator;  return !s.empty() && iterator == s.end();  }  void displaySchools(string path) {  ifstream myFile;  vector<string> temp1;  vector<string> temp2;  int temp;  myFile.open(path);  while (!myFile.eof()) {  readNextRow(myFile);    if (isDigit((data[8])) ){  temp = stoi(data[8]);  if (temp == 0) {  cout << "School Name: " << data[1] << endl;  cout<< "No. of students: " << temp << endl;  cout << endl;  }  }          }  }  void percentageSchools(string path) {  ifstream myFile;  myFile.open(path);  int noOfbigSchools=0;  int noOfschools = 0;  int temp;  while (!myFile.eof()) {  readNextRow(myFile);  if (isDigit((data[6]))) {  temp = stoi(data[6]);  if (temp >50) {    noOfbigSchools++;  }  noOfschools++;  }      }  cout << "Number of Schools: " << noOfschools << endl;  cout << "Number of big Schools : " << noOfbigSchools << endl;  double percentage = ((float)noOfbigSchools / (float)noOfschools) \* 100;  cout << "The Percentage of Schools are given as: " << percentage << "%" << endl;  }  };  int main() {  cout << "Enter the name of the file:" << endl;  string fileName;  getline(cin, fileName);  ifstream File;  CSVRow\* csv = new CSVRow();  File.open(fileName);  csv->readNextRow(File);  csv->percentageSchools(fileName);  } |

**Output:**

**Text

Description automatically generated**

**Task D:**

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

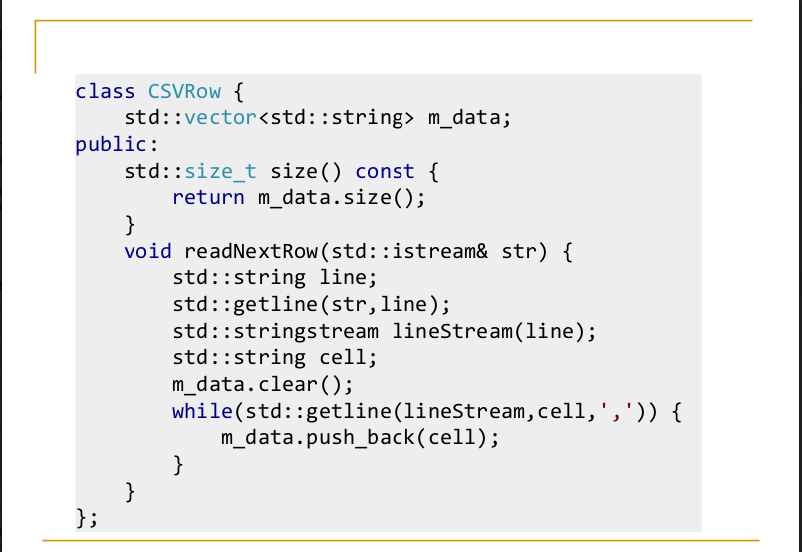
|  |
| --- |
| #include<iostream>  #include<vector>  #include<string>  #include<sstream>  #include<fstream>  using namespace std;  class CSVRow {  public:  vector<string> data;  size\_t size() const {  return data.size();  }  void readNextRow(istream& str) {  string line;  getline(str, line);  stringstream lineStream(line);  string cell;  data.clear();    while (getline(lineStream, cell, ','))  {  data.push\_back(cell);    }    }  void displayFile(string path) {  ifstream myFile;  myFile.open(path);  while (!myFile.eof()) {  readNextRow(myFile);  for (int i = 0; i < (int)size(); i++) {  cout << data[i];  }    cout << endl;  }      }  bool isDigit(const string& s)  { string::const\_iterator iterator= s.begin();  while (iterator != s.end() && isdigit(\*iterator))  ++iterator;  return !s.empty() && iterator == s.end();  }  void displaySchools(string path) {  ifstream myFile;  int temp;  myFile.open(path);  while (!myFile.eof()) {  readNextRow(myFile);    if (isDigit((data[8])) ){  temp = stoi(data[8]);  if (temp == 0) {  cout << "School Name: " << data[1] << endl;  cout<< "No. of students: " << temp << endl;  cout << endl;  }  }          }  }  void percentageSchools(string path) {  ifstream myFile;  myFile.open(path);  int noOfbigSchools=0;  int noOfschools = 0;  int temp;  while (!myFile.eof()) {  readNextRow(myFile);  if (isDigit((data[6]))) {  temp = stoi(data[6]);  if (temp >50) {    noOfbigSchools++;  }  noOfschools++;  }      }  cout << "Number of Schools: " << noOfschools << endl;  cout << "Number of big Schools : " << noOfbigSchools << endl;  double percentage = ((float)noOfbigSchools / (float)noOfschools) \* 100;  cout << "The Percentage of Schools are given as: " << percentage << "%" << endl;  }  void highestDropOut(string path) {  ifstream myFile;  myFile.open(path);  while (!myFile.eof()) {  readNextRow(myFile);    int temp;  float highDropOut=0.0;  string Name;  while (!myFile.eof()) {  readNextRow(myFile);  if (isDigit((data[6]))) {  temp = stoi(data[6]);  if (temp > 50 && stof(data[12]) > highDropOut) {  Name = data[1];  highDropOut = stof(data[12]);    }      }  }  cout << "School Name : " <<Name<< endl << "School Highest DropOut: "<<highDropOut << endl;  }  }  };  int main() {  cout << "Enter the name of the file:" << endl;  string fileName;  getline(cin, fileName);  ifstream File;  CSVRow\* csv = new CSVRow();  File.open(fileName);  csv->readNextRow(File);  csv->highestDropOut(fileName);  } |

**Output:**

Text

Description automatically generated

**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.