Data Structures – Spring 2023
Project 2
Due: February 27th 2023 at 11:59 PM

Objective:
The goal of this project is to create a tableClass class and perform queries in C++ as outlined in the project description.

Description:
In this project you will implement the tableClass class along with the all the methods that are given in the class definition (***boilerplate_project2.cpp***). You are required to read a csv file, do operations on the data stored and display the queried contents from the tableClass. TableClass is a table of data (all stored as strings) with rows and columns.

To perform operations on the data, there is something that we would need to discuss about. You can see in the class given that we are going to be storing the data (categorical and numeric) as strings. So, before we calculate the min of a given column, we need to check if that is a column whose values are in fact of a numeric data type for which we can calculate the requested statistic. And if it is a column of numbers for which we can calculate each of the requested statistic, we would have to convert the string to a numeric data type as and when we are using the value as a number for calculation purposes. This is referred to as **late binding** when we convert the stored data to a desired data type as and when required. The data types of each of the columns are also given in the

If asked to calculate min for a column that is not of a numeric data type, an **exception** is to be thrown. You are required to write an exception class and a set of exceptions for any and each of these cases. Please refer to the book for examples on exception throwing.

Input explanation:
The input file given is in the following format:

18 ← number of rows
6 ← number of columns
fileInput1.csv ← name of file input
string string char int int float ← data types of columns

F Ivan
F Ruth

V 42
V 124.9

D

I 4

C 0 3
C 2 5

R 5 10
R 12 15

S 0 2 0 5

The input file starts with the number of rows and then the number of columns in the file input document.
After this, we give you a list of commands/options for the operations and queries that go till the end of the input file. All row and column numbers are zero indexed.

- F – find the record/row and return the entire row. Query value will be the first columns value.
- V – find the value in the entire table and display all the row and column numbers where it appears.
- D – display the myTable data.
- I – find the min of the column number given.
- C – return a tableClass object with a subset of columns and all rows from the two given column numbers – Example: [0 3] => all columns between 0 and 3 inclusive.
- R – return a tableClass object with a subset of rows and all columns from the two given row numbers – Example: [10 15] => all rows between 10 and 15 inclusive.
- S – return a tableClass object with a subset of rows and columns from the two given row and column numbers – Example: [0 3 10 15] => all columns between 0 and 3 and rows between 10 and 15 inclusive.

The next place you'd have to throw an **exception** is when you are trying to search for a name that is not present in the dataframe.

Once you have read the file input document, you are **required to sort the table** with respect to the first column. You have to sort the table retaining the information in each of the rows – as in, moving entire rows while sorting as per the first column. All queries will be done only after you have sorted the table.

You are also required to write the method to overload the [] operator to query an entire row with just the index of the row.

To get you all started, a **boiler plate file** is given as well. This file is also given alongside this project. Keep up with the FAQ discussion for the project as more assumptions will be updated there. Sample output file will be released soon as well.

CSV file:

The name of the csv file to open and read is given in the input file above. You will use the fstream functionalities to open and read the contents of the csv file. Read in all the data from the

csv file and store it in the 2D string array called myTable in the tableClass class. Notice again that the data can be of numbers or names, they all have to be stored as a string. Below is an example table of information that will be read from the csv file.

Dave	Philadelphia	M	39	72	167.6
Carl	Izmir	M	32	70	155.9
Alex	Singapore	M	41	74	170.5
Bert	Zhongshan	M	42	68	166.8
Luke	Porto Alegre	M	34	72	163.6
Myra	Karaj	F	23	62	98.8
Elly	Vienna	F	30	66	124.9
Jake	Ulsan	M	32	69	143.5
Fran	Hamburg	F	33	66	115.5
Omar	Kampala	M	38	70	145.4
Page	Tehran	F	31	67	135.2
Quin	Chennai	M	29	71	176.0
Hank	Shanghai	M	30	71	158.7
Ivan	London	M	53	72	175.9
Kate	Patna	F	47	69	139.3
Neil	Daejeon	M	36	75	160.9
Ruth	Managua	F	28	65	131.8
Gwen	Bucharest	F	26	64	121.1

There are 6 columns and 18 rows in the above table. All this information will have to be stored as strings in the myTable 2D array in class tableClass.

Redirected input:

Redirected input provides you a way to send a file to the standard input of a program without typing it using the keyboard. To use redirected input in Visual Studio environment (on windows), follow these steps: After you have opened or created a new project, on the menu go to project, project properties, expand configuration properties until you see Debugging, on the right you will see a set of options, and in the command arguments type <"input filename". The < sign is for redirected input and the input filename is the name of the input file (including the path if not in the working directory).

If you use macOS or linux (or windows using powershell), you may use the terminal to compile and run your program using g++. This is the compiler that we use in the autograder on GradeScope (you may need to install g++ on your computer). Once you installed g++, you may compile your program using the following command:
g++ project2.cpp -o p1

You may then run the program with the redirected input using the following command:
./p1 < input1.txt

For powershell,

Get-Content input1.txt | ./p1

Make sure your program and your input file are in the same folder.

You may also use Google CoLab as an alternative and a sample file is given along with the project. This is an ipynb file which has to be opened in Google CoLab.

A simple program that reads the input file (one number) and displays everything that was read is given below.

```
#include<iostream>
using namespace std;

int main()
{
  int num = 0;
  cout << "Hello World" << endl;
  cin >> num;
  cout << num;
  return 0;
}
```

Sample output file for corresponding input files will be released. The input1.txt file given is a simple input file that will help you understand the project, more complicated ones will be released later and used for grading.

Submission:

Submission will be through GradeScope. Your program will be autograded with test cases and then hand graded to check for documentation and if you have followed the specifications given. You may submit how many ever times to check if your program passes the test cases provided. Test cases will be released at the beginning and later other test cases will be released while grading. For the autograder to work, the program you upload must be named as **project2.cpp**. This will be the only file you will submit. Make sure it is well commented.

Constraints:

1. In this project, the only header you will use the header files that are given to you in the boiler plate code. Using other or excess header files will be subject to a heavy grade penalty for the project.
2. None of the projects is a group project. Consulting with other members of this class on programming projects is strictly not allowed and plagiarism charges will be imposed on students who do not follow this.
    - Please review academic integrity policies in the modules.

How to ask for help:

1. Check FAQ discussion board of the project first.
2. Email the TAs with your precise questions.

3. Upload your well commented and clean program to CodePost.
    a. This is a website which is used to share code with your TA only.
    b. You will upload your program and I can view it and comment on it.
    c. Here is the invite link to our class for the summer.
    d. Once you join the class on CodePost, you can upload your program to the Project 1 assignment.

        Note: Your program will **not** be auto graded at CodePost, this is just for when you want to ask a question and a place where I can look at your program and comment on it. CodePost is great for live feedback. GradeScope is the place where you should submit and where your program will be autograded.