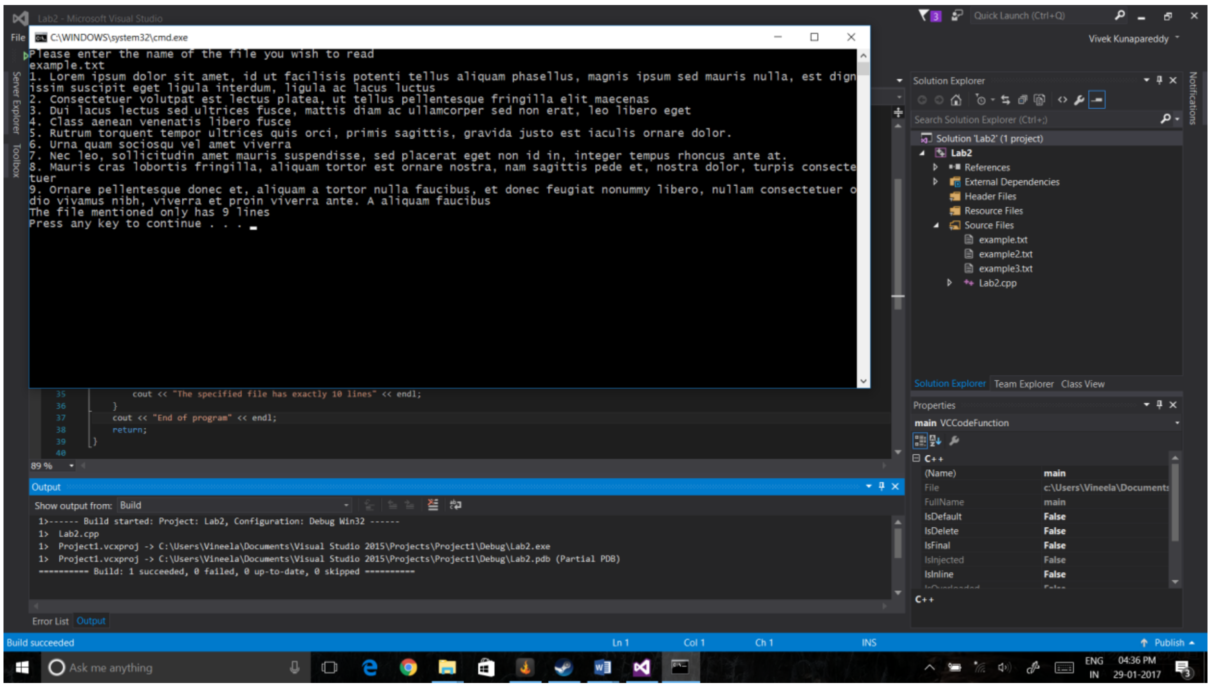
Data Structures Lab 2 Report

Group Members: Vivek Kunapareddy, Yuan Cheng

a) The concepts explored in this lab were: File I/O and Object oriented programming(Class design and instantiating objects)
These concepts are important in this class as most of data structures will involve some sort of OOP concept or class design. This assignment helps us get a strong grip on fundamentals for when we will be experimenting later on in this course.
These concepts will be extremely helpful when moving into the workforce as most modern programming involves OOP and File management especially in a field such as Embedded Systems.

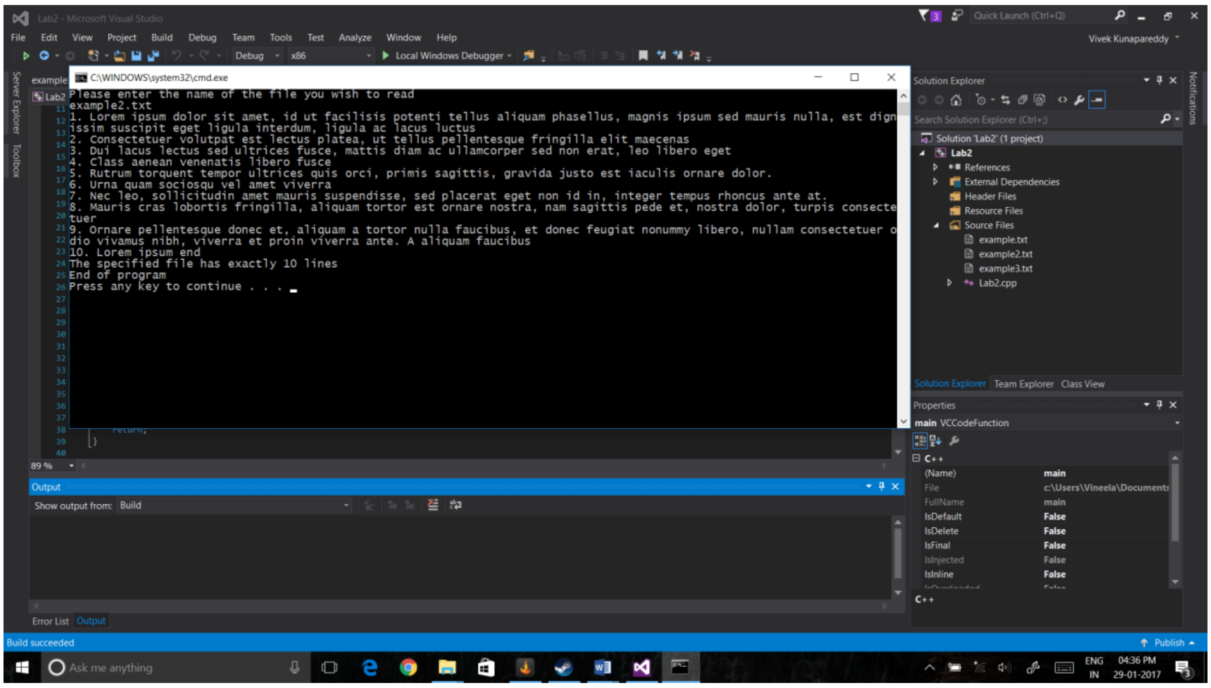


```
Please enter the name of the file you wish to read
example.txt
1. Lorem ipsum dolor sit amet, id ut facilisis potenti tellus aliquam phasellus, magnis ipsum sed mauris nulla, est dign
issim suscipit eget ligula interdum, ligula ac lacus luctus
2. Consectetuer volutpat est lectus platea, ut tellus pellentesque fringilla elit maecenas
3. Dui lacus lectus sed ultrices fusce, mattis diam ac ullamcorper sed non erat, leo libero eget
4. Class aenean venenatis libero fusce
5. Rutrum torquent tempor ultrices quis orci, primis sagittis, gravida justo est iaculis ornare dolor.
6. Urna quam sociosqu vel amet viverra
7. Nec leo, sollicitudin amet mauris suspendisse, sed placerat eget non id in, integer tempus rhoncus ante at.
8. Mauris cras lobortis fringilla, aliquam tortor est ornare nostra, nam sagittis pede et, nostra dolor, turpis consecte
tuer
9. Ornare pellentesque donec et, aliquam a tortor nulla faucibus, et donec feugiat nonummy libero, nullam consectetuer o
dio vivamus nibh, viverra et proin viverra ante. A aliquam faucibus
The file mentioned only has 9 lines
Press any key to continue . . .
```

```
            cout << "The specified file has exactly 10 lines" << endl;
        }
        cout << "End of program" << endl;
        return;
}
```

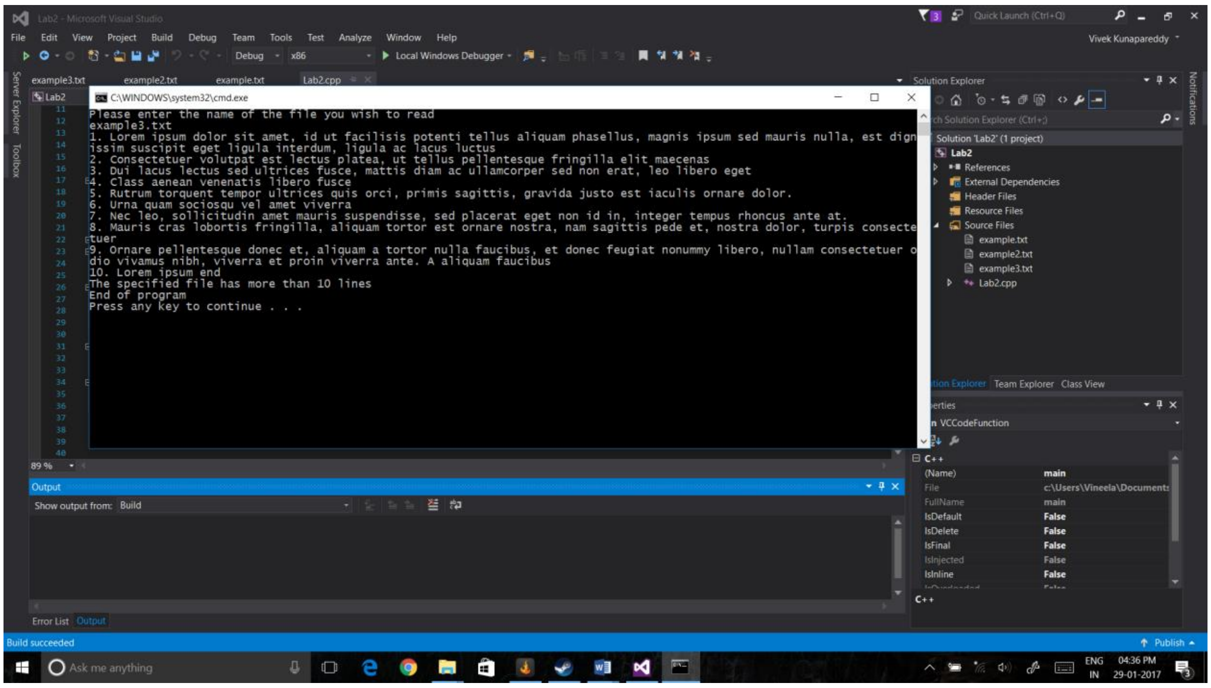
```
1>------ Build started: Project: Lab2, Configuration: Debug Win32 ------
1>  Lab2.cpp
1>  Project1.vcxproj -> C:\Users\Vineela\Documents\Visual Studio 2015\Projects\Project1\Debug\Lab2.exe
1>  Project1.vcxproj -> C:\Users\Vineela\Documents\Visual Studio 2015\Projects\Project1\Debug\Lab2.pdb (Partial PDB)
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Task 1 – Output with 9 Lines



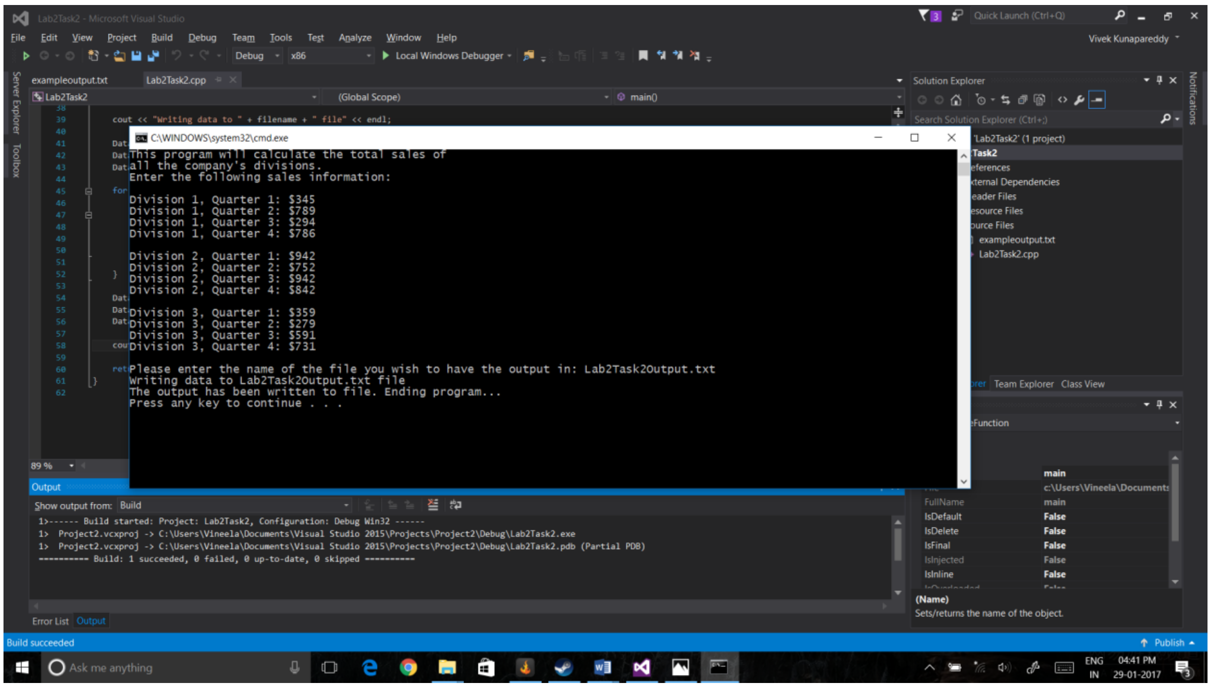
```
Please enter the name of the file you wish to read
example2.txt
1. Lorem ipsum dolor sit amet, id ut facilisis potenti tellus aliquam phasellus, magnis ipsum sed mauris nulla, est dign
issim suscipit eget ligula interdum, ligula ac lacus luctus
2. Consectetuer volutpat est lectus platea, ut tellus pellentesque fringilla elit maecenas
3. Dui lacus lectus sed ultrices fusce, mattis diam ac ullamcorper sed non erat, leo libero eget
4. Class aenean venenatis libero fusce
5. Rutrum torquent tempor ultrices quis orci, primis sagittis, gravida justo est iaculis ornare dolor.
6. Urna quam sociosqu vel amet viverra
7. Nec leo, sollicitudin amet mauris suspendisse, sed placerat eget non id in, integer tempus rhoncus ante at.
8. Mauris cras lobortis fringilla, aliquam tortor est ornare nostra, nam sagittis pede et, nostra dolor, turpis consecte
tuer
9. Ornare pellentesque donec et, aliquam a tortor nulla faucibus, et donec feugiat nonummy libero, nullam consectetuer o
dio vivamus nibh, viverra et proin viverra ante. A aliquam faucibus
10. Lorem ipsum end
The specified file has exactly 10 lines
End of program
Press any key to continue . . .
```

Task 1 – Output with 10 lines



```
Please enter the name of the file you wish to read
example3.txt
1. Lorem ipsum dolor sit amet, id ut facilisis potenti tellus aliquam phasellus, magnis ipsum sed mauris nulla, est dign
issim suscipit eget ligula interdum, ligula ac lacus luctus
2. Consectetuer volutpat est lectus platea, ut tellus pellentesque fringilla elit maecenas
3. Dui lacus lectus sed ultrices fusce, mattis diam ac ullamcorper sed non erat, leo libero eget
4. Class aenean venenatis libero fusce
5. Rutrum torquent tempor ultrices quis orci, primis sagittis, gravida justo est iaculis ornare dolor.
6. Urna quam sociosqu vel amet viverra
7. Nec leo, sollicitudin amet mauris suspendisse, sed placerat eget non id in, integer tempus rhoncus ante at.
8. Mauris cras lobortis fringilla, aliquam tortor est ornare nostra, nam sagittis pede et, nostra dolor, turpis consecte
tuer
9. Ornare pellentesque donec et, aliquam a tortor nulla faucibus, et donec feugiat nonummy libero, nullam consectetuer o
dio vivamus nibh, viverra et proin viverra ante. A aliquam faucibus
10. Lorem ipsum end
The specified file has more than 10 lines
End of program
Press any key to continue . . .
```

Task 1 – Output with 11 Lines



```
cout << "Writing data to " + filename + " file" << endl;
```

```
This program will calculate the total sales of
all the company's divisions.
Enter the following sales information:

Division 1, Quarter 1: $345
Division 1, Quarter 2: $789
Division 1, Quarter 3: $294
Division 1, Quarter 4: $786

Division 2, Quarter 1: $942
Division 2, Quarter 2: $752
Division 2, Quarter 3: $942
Division 2, Quarter 4: $842

Division 3, Quarter 1: $359
Division 3, Quarter 2: $279
Division 3, Quarter 3: $591
Division 3, Quarter 4: $731

Please enter the name of the file you wish to have the output in: Lab2Task2Output.txt
Writing data to Lab2Task2Output.txt file
The output has been written to file. Ending program...
Press any key to continue . . .
```

```
1>------ Build started: Project: Lab2Task2, Configuration: Debug Win32 ------
1>  Project2.vcxproj -> C:\Users\Vineela\Documents\Visual Studio 2015\Projects\Project2\Debug\Lab2Task2.exe
1>  Project2.vcxproj -> C:\Users\Vineela\Documents\Visual Studio 2015\Projects\Project2\Debug\Lab2Task2.pdb (Partial PDB)
========== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped ==========
```

Lab2Task2 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug x86 Local Windows Debugger
exampleoutput.txt Lab2Task2.cpp
(Global Scope) main()
Solution Explorer
Output
Show output from: Build
Error List Output
Build succeeded

Task 2 - Output

b) The file access flags used for the most part were ios::app as this allowed to use a single flag to both create non existing files and also append output to existing files. The ios::in flag was used to read input from the file in task 3 too. The ios:: binary flag was disregarded for this lab as we didn't need to deal with binary data. The ios::trunc and ios::out flags were also not used as they delete existing data in the file which was not required.

c) The class design was extremely simple in Task 3 as it just includes 4 private member variables pertaining to information about the product. And the public member functions include getters and setters for each member variable. The reason for the simple design was to encase the logic used in the main file instead of in the class. Thus making them reusable. Also the getters and setters were included to allow access to the member variables.

d) The testing in Task 3 was done by checking if the file inputted existed, and then counting the number of lines in the file. This was done as the number of products was fixed beforehand. If the number of products was lesser or more than the required number, an error message will be outputted to the console. The test cases were considered before writing the code as they were specified in the requirements

Special Instructions:

All the file names should be entered as fullname.extension. Eg: "Lab2 – Task1 Input9Lines"

For testing output to files which need to be created, please include extensions when naming the file. Eg: "Lab2Output.txt"

Group Contributions:

The entire programming was done as a pair during the lab

The class design was done by Yuan

The file I/O logic was handled by Vivek

The debugging was done together while pair programming