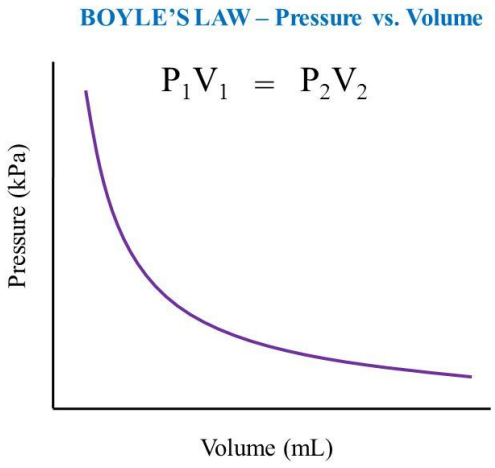
Homework Assignment #5

GE 1501
Cornerstone of Engineering 1
CRN: 16504
Monday, Wednesday, Thursday from 10:30 AM - 11:35 AM
429 Dana Research Center
Wednesday, October 4, 2017
Jordan Lian

Table of Contents



Reading a Set of Grades

Pseudocode/Flowchart

1. Declare variables
2. Setup file pointers
    a. Ofstream, and ifstream
3. Set value of file pointers
4. Use a while loop to continuously keep adding the number of letter grades until there is no more
5. Use case tests to add up characters by 1 for each letter
6. Print the results using cout

Test by hand calculation

Count letters by hand

21 As, 16 Bs, 18 Cs, 5 Ds, 8 Fs

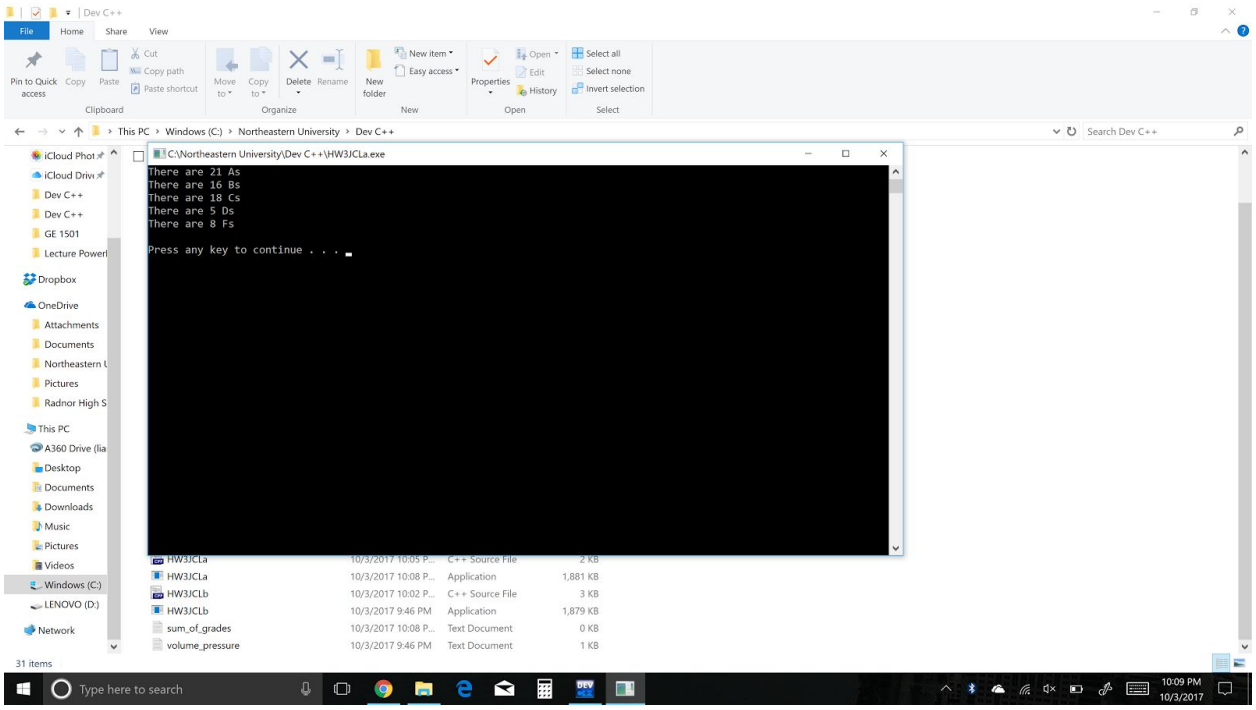
Program source code .cpp file

```
//This program will read in a set of grades from a file called "grade.txt"
//This program will report the total number of letter grades (A, B, C, D, F) in the file
//This program will use logic and the switch statement

//libraries

#include <iostream> //almost always include iostream
#include <fstream> //need ifstream and ofstream to set up file pointer permission (o = output, i = input)
#include <string> //c_str used with .open() and allows for user input of a file name

using namespace std;

int main(void)
{
	//declare variables
	char grade; //grade character
	int A = 0, B = 0, C = 0, D = 0, E = 0, F = 0; // initialization of variables
	string filename = "grade.txt"; //input filename
```

```
	//set up file pointers and permission to read or write
	ofstream outfile; //outfile is a variable - permission to write
	ifstream infile; //infile is a variable - permission to read

	//set value of file pointer to file names using .open()
	outfile.open("sum_of_grades.txt"); //write output to this file)
	infile.open(filename.c_str()); //use function c_str so that .open() can see the value of
file_name

	while (infile >> grade) { //use a while loop get the number of each letter grades

	switch (grade)

	{ //set up switch to test input

	case 'A': //set up case to test if true
		A++;
		break; //end switch test since true

	case 'B': //set up case to test if true
		B++;
		break; //end switch test since true

	case 'C': //set up case to test if true
		C++;
		break; //end switch test since true

	case 'D': //set up case to test if true
		D++;
		break; //end switch test since true

	case 'F': //set up case to test if true
		F++;
		break; //end switch test since true

	}
}

	cout << "There are " << A << " As\n";
	cout << "There are " << B << " Bs\n";
	cout << "There are " << C << " Cs\n";
	cout << "There are " << D << " Ds\n";
```

```
        cout << "There are " << F << " Fs\n\n";

        system("pause");
        return 0;
}
```

Screenshot of Output



Volume vs Pressure

Pseudocode/Flowchart

1. Define constants
    a. a, b, R
2. Declare variables
    a. T, n, V_0, V_f, V_i (increment), P
3. Outfiles to store the calculated data
4. Enter inputs
    a. Enter the initial/final/increment volumes (V_0, V_f, V_i respectively), temperature (T), moles (n)
5. Print entered values again using cout
6. Implement a for loop to add the incremental volume values and the corresponding calculated pressures

a. Enter the formula for pressure into the for loop
b. Don't forget to include outfile for all of the printed results

Test by hand calculation

$P = [nRT / (V/1000 - bn)] - (an^2 / (V^2/1000^2))$

Sample Calc

V = 400 mL
T = 300 K
n = 0.02 mol CO_2
a = 3.592 L^2*atm / mol^2
b = 0.0427 L / mol for CO_2
R = Gas Constant = 0.08206 (L*atm) / (mol*K)

[0.02 ~~mol~~ * 0.08206 (~~L~~*atm) / (~~mol~~*~~K~~) * 300 ~~K~~] / [400 ~~mL~~ * (1 ~~L~~ / 1000 ~~mL~~) - 0.0427 ~~L~~ / ~~mol~~ * 0.02 ~~mol~~] - (3.592 ~~L^2~~*atm / ~~mol^2~~ * 0.02 ~~mol~~)^2 / [(400 ~~mL~~)^2 (1 ~~L~~ / 1000 ~~mL~~)^2]) = **1.2246 atm**

Program source code .cpp file

```
/* This program will calculate the pressure of carbon dioxide given the volumes input into the program
We will use the Van der Waals equation to calculate this
There will be a table that illustrates the relationship between pressure, volume, and temperature
The table values will be based on the volume increment */

//libraries

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

//define constants

#define R 0.08206 // units of (L * atm) / (mol * K)
#define a 3.592 // units of L^2 * atm / mol^2 for carbon dioxide
#define b 0.0427 // units L / mol for carbon dioxide

//declare variables

double n;
double T;
```

```
double V0;
double V1;
double VI;
double P;

int main (void)

{

        //outfiles to store output data
        ofstream outfile;
        outfile.open("volume_pressure.txt");

        //enter inputs

        cout << "Please enter the amount of carbon dioxide in moles (mol) --> ";
        cin >> n;

        cout << "Please enter the temperature in Kelvin (K) --> ";
        cin >> T;

        cout << "Please enter the initial volume in milliliters (mL) --> ";
        cin >> V0;

        cout << "Please enter the final volume in milliliters (mL) --> ";
        cin >> V1;

        cout << "Please enter the volume increment in milliliters (mL) --> ";
        cin >> VI;

        //show values entered in the program (basically a repeat and confirmation)

        cout << "\nQuantity of carbon dioxide (moles) > " << n;
        outfile << "Quantity of carbon dioxide (moles) > " << n;

        cout << "\nTemperature (K) > " << T;
        outfile << "\nTemperature (K) > " << T;

        cout << "\nInitial Volume (milliliters) > " << V0;
        outfile << "\nInitial Volume (milliliters) > " << V0;

        cout << "\nFinal Volume (milliliters) > " << V1;
        outfile << "\nFinal Volume (milliliters) > " << V1;
```

```
        cout << "\nVolume Increment (milliliters) > " << VI << "\n";
        outfile << "\nVolume Increment (milliliters) > " << VI << "\n";

        cout << "\n" << n << " moles of carbon dioxide at " << T << " Kelvin\n\n";
        outfile << "\n" << n << " moles of carbon dioxide at " << T << " Kelvin\n\n";

        cout << "Volume (mL) \t Pressure (atm)\n\n"; //organizing the tables
        outfile << "Volume (mL) \t Pressure (atm)\n\n";

        for (double x = V0; x <= V1; x += VI) //use to keep adding the increment amount to the original volume until the sum is equal to the final volume

        {
                P = n*R*T / (x/1000 - b*n) - (a * n * n  / (x*x/1000/1000)); //solving for pressure here

                cout << x << " \t\t" << P << "\n"; //volume to the left, pressure to the right
                outfile << x << " \t\t" << P << "\n";
        }

system ("pause");
return 0;

        }
```

Screenshot of Output