CSCI312 - Character Counter - Array

Design a program that reads an ASCII text file (provided) one byte at a time and produces an output file that contains the frequency count of how many times each character appears in the input file. Do not sort the output. A character frequency class instance must be used to represent each unique character in the file (see below). For this exercise, the character frequency objects **must** be processed **and** stored using an array (no other data structure is acceptable).

For example, given the sample input file:

Hello.

Would yield the following output file.

```
 (10)     1
 (13)     1
.(46)     1
H(72)     1
e(101)    1
l(108)    2
o(111)    1
```

The output file must follow the format presented above. Character (ASCII Value) (tab) Frequency. Note that formatting anomalies are expected due to the nature of the characters themselves (e.g., the tab character, carriage return, etc.) and will not cause a point deduction. You may elect to filter out characters that did not appear in the input file (zero count).

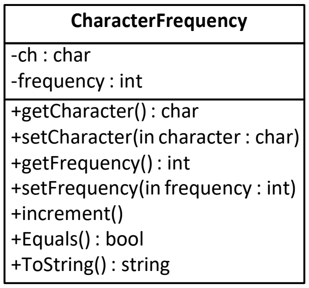
The program should be able to be executed from the command line using the following syntax: programname.exe <inFile> <outFile> (**optional for this assignment**).

The program must accept the full path (including the exact filename) as input. Do not sandbox, alter the input parameter, or require specific filenames or extensions.

For example, counter.exe myInput.txt Count.txt

Where counter.exe is the program name, myInput.txt is the input ASCII file, and the Count.txt is the output file.

The character frequency class must implement the essence of the object described in the UML diagram below. Note that methods described in the UML Class diagram below are not absolute and may be changed as desired as long as the basic functionality is preserved.



CharacterFrequency

-ch : char
-frequency : int

+getCharacter() : char
+setCharacter(in character : char)
+getFrequency() : int
+setFrequency(in frequency : int)
+increment()
+Equals() : bool
+ToString() : string

You may use C++, VB, C#, or Java to implement this program as long as the following requirements are met. A C++, VB, or C# project must be created using Visual Studio 2019. Java programs must be created in the Eclipse project format or provided in source code form with instructions on how to compile and run the program. Regardless of the language used, the entire project must be submitted as a single ZIP file that contains the project folder, source code, and documentation.

Upload your completed project to Blackboard as a single ZIP or 7z file.

Grading Rubric

Array-based character frequency counter	**Possible**	**Score**	**Comment**
The Command Line Interface (CLI) is utilized to obtain program parameters (optional)	0		+ 5 extra credit
Character Frequency Class Exists	10	✗	
The ToString() method is overloaded and produces output using properties of the Character Frequency class.	10	✗	
The Equals() method is overloaded	20	✗	
The Equals method returns false when null or a non-Character Frequency object instance is passed	10	✗	
The Equals method returns true when the same object instance is passed	5	✗	
Character Frequency Class is used in conjunction with an **Array** to Count the Characters in the input file. The use of any other data structure is disallowed.	25	✗	
The input file is read byte-by-byte (File.ReadAllText, ReadLine, or other similar mass reading methods are disallowed). The use of the BinaryReader (C# or VB), BufferedInputStream (Java), or ifstream (in ios::binary mode for C++) is required.	25	✗	
An output file is created	10		
Output format matches stated requirements	20	✗	
Output counts are accurate	35	✗	
The program does not crash due to input/output errors	5		
Total (note the percentage is used in Blackboard not the points)	**175**		