National University of Computer and Emerging Sciences, Lahore Campus

Course:	OOP (Lab)	Course Code:	CL 217
Program:	BS (Computer Science)	Semester:	Fall 2019
Duration:	2 Hours	Total Marks:	60
Paper Date:	12-Oct-2019	Weight	30%
Section:	All	Page(s):	3
Exam:	Lab Midterm Exam	Reg. No	

Important Instructions (Please read them before attempting the exam):

- Submit **ONLY .cpp File** in this format (Make the File **named** with your **Roll Number** e.g., L13-4152).
- **Plagiarism** will result in **F grade** in lab.
- No cell phones are allowed. Sharing of **USBs** or any other items is **not allowed.**
- Submission path will be announced soon.
- Necessary files are placed on **\\cactus\Xeon\Fall 2019\Shakeel Zafar\OOP MID Exam\Files**
- Use **Visual Studio 2012.**

Question :

You are given four text files, your task is to compute the probabilities of each word from the corpus/data. The "data.txt" can have any type of text in it, we only want to compute the probability of the words that have English alphabets only.

Probability can be computed as:

$$\text{Probability (word)} = \frac{count(word)}{Total\ Words}$$

Sample **"data.txt":**

Hello to 3v3ry Body. Yours's sincerely. hello I am "Ali". I AM no body.

Total words: 13

The above results obtained using these three filter conditions.

Word	Count	Probability
hello	2	2/13
to	1	1/13
body	2	2/13
yours	1	1/13
sincerely	1	1/13
i	2	2/13
am	2	2/13
ali	1	1/13
no	1	1/13

1. Convert every word to lowercase.
2. Ignore the words with numeric character/s.
3. Remove special characters (e.g., ' , " " .) it means remove every character other than English alphabets.

Use this class to fulfill the above conditions for the word read from the file.

```
class Helper
{
public:
	static char* filterStr(char* str) {
		// This function receives a word's text and returns a filtered text
	}
	static char* GetStringFromBuffer(char* str)
	{
		//This function should allocate required memory on heap,
		//copy string in this memory and return newly created cstring.
	}
};
```

To keep record of word and its count you are required to design a class **"Word",** that has three data members. You need to implement all the necessary members (accessors/mutators) functions of class "Word" to execute the program successfully.

```
class Word {
	char * text;
	int count; float probability;
public:
	void assign(Word obj);
	void printWord();
	//member functions | decide yourself
};
```

Implement a global function ***getWordsList*** that receives text file name and returns **dynamically allocated Word Array**. This function opens the file, reads the data, apply filters on data and store in dynamic array of Word. There shouldn't be memory leakage. Make this dynamically array of size ONE initially. Whenever you find a new word, extend this array to SIZE + 1 and store the words.

Use the member function ***assign,*** which assigns the values of right-hand object to this object.

Implement a member function ***printWord*** *of class* ***Word*** which displays the **text** and its **probability**.

```
Word *getWordsList(char *filename, int & DistinctWords);
void main() {
	const int filesCount = 4;			//maximum files
	int totalWords[filesCount];			//total words in each file
	Word** list = new Word*[filesCount];	//List of pointers to hold the words/file
	char fileName[filesCount][10] = { "data1.txt","data2.txt", "data3.txt", "data4.txt" };
	//Names of files
	for (int i = 0; i<filesCount; i++)		//For every file
	list[i] = getWordsList(fileName[i], totalWords[i]); //find words and probabilities
	for (int i = 0; i<filesCount; i++)		//for every file
		for (int j = 0; j < totalWords[i]; j++)	//for every word in file-i
			list[i][j].printWord();		//print the word and its probabilities
	//There should not be any memory leackage

}
```