

Programming Challenges using loops: File and character processing - (WEEK#9)

Programming Challenge#1: Write a program that will open and read character-by-character the contents in file history.dl. Use appropriate validation for file processing. How many nonblank characters are there in file [history.dl](#)? Add the necessary statements including a counter to your program that keeps track of the number of nonblank characters in the file. Print this number to the screen. Do not include '\n' in your nonblank count.

Programming Challenge#2: Write a functional design and a program to analyze a sample of text in a file.

Count the instances of the following categories of symbols:

- Uppercase letters
- Lowercase letters
- Digits
- End-of-sentence markers (periods, explanation points, and question marks)
- Intrasentence markers (commas, semicolons, and colons)
- Blanks
- All other symbols

Use a switch statement in your processing.

After collecting these statistics, use them to approximate the following statistics:

- Average word length
- Average sentence length

Design and implement a test plan for the program.

A sample program showing file read operations ...

```
//*****
// EchoLine program
// This program reads and echoes the characters from one line
// of an input file
//*****
#include <iostream>
#include <fstream>           // For file I/O

using namespace std;

int main()
{
    char    inChar;           // An input character
    ifstream inFile;         // Data file

    inFile.open("text.dat");  // Attempt to open input file
    if (!inFile)             // Was it opened?
    {
        cout << "Can't open the input file."; // No--print message
        return 1;             // Terminate program
    }
    inFile.get(inChar);       // Get first character
    while (inChar != '\n')
    {
        cout << inChar;       // Echo it
        inFile.get(inChar);    // Get next character
    }
    cout << endl;
    return 0;
}

/*datafile contents ...
Now is the time for all good men to come to the aid of their party.
*/
```

You may also find the CharCounts.cpp below helpful.

```
//*****
// CharCounts program
// This program counts the number of letters, periods, question
// marks, and exclamation marks found in the first 100 input
// characters
// Assumption: Input consists of at least 100 characters
//*****
#include <iostream>
#include <cctype>           // For isalpha()

using namespace std;

int main()
```

```

{
    char inChar;           // Current input character
    int loopCount;         // Loop control variable
    int letterCount = 0;   // Number of letters
    int periodCount = 0;   // Number of periods
    int questCount = 0;    // Number of question marks
    int exclamCount = 0;   // Number of exclamation marks

    cout << "Enter your text:" << endl;
    for (loopCount = 1; loopCount <= 100; loopCount++)
    {
        cin.get(inChar);
        if (isalpha(inChar))
            letterCount++;
        else
            switch (inChar)
            {
                case '.' : periodCount++;
                           break;
                case '?' : questCount++;
                           break;
                case '!' : exclamCount++;
                           break;
                default  : ;    // Unnecessary, but OK
            }
    }
    cout << endl;
    cout << "Input contained" << endl
        << letterCount << " letters" << endl
        << periodCount << " periods" << endl
        << questCount << " question marks" << endl
        << exclamCount << " exclamation marks" << endl;
    return 0;
}

```

```

/word_char_processing.cpp
#include<iostream>
#include<cctype>
using namespace std;

int main()
{
    string word;
    char letter;
    cout<<"Type a single word, press enter key: ";
    cin>>word;
    cout<<word<<endl;

    cout<<word.length()<<endl;
    cout<<word[0]<<endl;
    letter=word[0];
    cout<<letter<<endl;
    letter=tolower(letter);
    cout<<letter<<endl;

    for(int count=0; count<word.length();count++)
    {
        letter=toupper(word[count]);
        cout<<letter;
    }

    cout<<endl;
    for(int count=0; count<word.length();count++)
    {
        letter=toupper(word[count]);
        switch(letter)
        {
            case 'A':cout<<"apple ";
                     break;
            case 'n': cout<<"novel ";
                     break;
            case 't': cout<<"trial ";
                     break;
        }
    }
    return 0;
}

```