Programming Fundamentals

|  |  |
| --- | --- |
| **Lab-3** | |
| **Topic** | Basics of file handling and character array |
| **Objective** | Learning objectives of this lab are to implement code related to basics of character array and file handling . |

# Task 1:

Perform the following task:

* Create a file **“sample.txt”.**
* Ask user to enter integers 1-10 one by one.
* Now enter those integers to the “sample.txt” file.

**Outpu**t: 1 2 3 4 5 6 7 8 9 10

# Task 2:

Perform the following task:

* Create a file **“integer.txt”.**
* Insert data in this file “21325”.
* Create a file **“character.txt”.**
* Insert data in this file “L”.
* Create a file **“characterArray.txt”.**
* Insert data in this file “Programming Fundamentals”.

# Task 3:

Perform the following tasks:

* Read an integer from file **“integer.txt”** and display on console.

|  |  |  |  |
| --- | --- | --- | --- |
| integer.txt |  |  | Console |
| 21325 |  |  | 21325 |

* Read a char from file **“character.txt”** and display on console.

|  |  |  |
| --- | --- | --- |
| character.txt |  | Console |
| L |  | L |

* Read a char array from file **“characterArray.txt”** and display on console.

|  |  |  |
| --- | --- | --- |
| characterArray.txt |  | Console |
| Programming Fundamentals |  | Programming Fundamentals |

* Repeat above task, and store data in **“NewFile.txt”** (using **out** mode)**.**

|  |  |  |
| --- | --- | --- |
| integer.txt |  | NewFile.txt |
| 21325 |  | 21325 |

|  |  |  |
| --- | --- | --- |
| character.txt |  | NewFile.txt |
| L |  | L |

|  |  |  |
| --- | --- | --- |
| characterArray.txt |  | NewFile.txt |
| Programming Fundamentals |  | Programming Fundamentals |

# Task 4:

Write a program that reads 5 integer numbers from a file **“integer.txt”,** and store sum and average in

**“result.txt”.**

|  |  |
| --- | --- |
| integer.txt | Result.txt |

|  |  |  |
| --- | --- | --- |
| 2 1 3 2 5 |  | 13 //sum  2.6 //Average |

Task 5:

Write a program that reads an average from file **“result.txt”,** and store the status in **“newfile.txt”** on the basis of following criteria:

**If average < 4, then below average**

**If 4 <= Average <= 6, then on average If average > 6, then above average**

|  |  |  |
| --- | --- | --- |
| average.txt |  | newfile.txt |
| 1 |  | below average |

# Task 6:

Write program that reads count from file, which shows how many names are there in file. Read those names from file **“data.txt”**, and store them in **“names.txt”** without count of names.

|  |  |  |
| --- | --- | --- |
| data.txt |  | names.txt |
| 4  lina  Amber Amna Shoaib |  | Lina  Amber Amna Shoaib | |

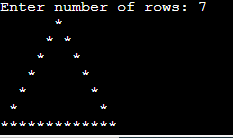
# Task 7:

Write program that gets a number ‘n’ from console, and display the pyramid of ‘\*’of that number ‘n’ on console as well as file ”shapes.txt”.

Input:

Enter number of rows : 7

Output: On console and in file(shapes.txt)



# Task 8:

Write a C++ program that creates a char array of size 10, Read a word containing lowercase,uppercase,symbols and integers form file **data.txt** . Count all the lowercase alphabets /upper case alphabets/digits/symbols/vowels and store the count of each in **result.txt.**

**Expected Output in File(Result.txt) :**Input From File (**data.txt**)(numbers , letters, symbols): cV?789kieQ  
Lowercase Alphabet count: 4  
Uppercase Alphabet count: 2  
Digits count: 3  
Symbols Count: 1  
Vowel Count: 3

# Task 9:

Write a program in which, you read a file **“sentence.txt”** into a character array. Now separate each word and store the word into a file **“word.txt”.**

|  |  |  |
| --- | --- | --- |
| **sentence.txt** |  | **word.txt** |
| Good Day Sir | Good  Day  Sir |

# Task 10:

Write a program in which, you read a file **“sentence.txt”** into a character array. Now separate each word and find its length now store the word and its length into a file **“word.txt”.**

|  |  |  |
| --- | --- | --- |
| **sentence.txt** |  | **word.txt** |
| Good Day Sir | Good 4  Day 3  Sir 3 |