Practical No. 05

Aim: Implement data cleansing and applying uppercase of first name and last name on C++.

Practical No. 05

Aim: Implement data clean~~ing~~sing ~~apply~~ and applying uppercase of first name and last name on C++.

Theory: Data cleaning is utmost important for any kind of data before you can make sense out of it and generate reports.

Let's say you get a ton of data which is extremely inconsistent in spaces, Mumbai is spelt as "Mumbai" , " Mum bai" , " Mumbai" , " Mumbai ". All these ~~six~~ four different spellings of Mumbai will be treated as four different cities , which will ultimately lead to you taking the wrong business decisions if you don't realise.

Data cleansing or cleaning is the process of identifying and removing (or correcting) inaccurate records from a dataset , table or database and refers to recognising unfinished, unreliable , inaccurate or non relevant parts of the data and then restoring, remodelling , or removing the dirty or crude data.

Data clearing in excel:

In excel, we have a lot of functions to do this kind of cleanup of spaces and many more data cleaning ~~te~~ techniques. Some of the functions are:

=TRIM is used for removing extra spaces

=CLEAN removes all non printable characters from cell

=UPPER used to convert all characters to uppercase

=LOWER used to convert all characters to lowercase

=PROPER used to convert first character to uppercase and others to lowercase.

Data cleaning examples and Data Cleaning Methods in Excel .

Get rid of extra spaces

Here, I have the text Welcome To Digital Vidya written in four different ways.

welcome to digital vidya
welcome   to digital vidya
welcome   to digital vidya
welcome   to   digital   vidya.

So, to clean this data and get rid of these extra spaces you can use the function 'trim'

Syntax: TRIM (TEXT)

Change text to Lower/Upper/Proper case:

Here, I have names written in different ways you can see either it could be all caps, it could be all lowercase and in some cases, it's a mix and match of uppercase, lowercase so to make it all consistent you can use one of the following formulas:

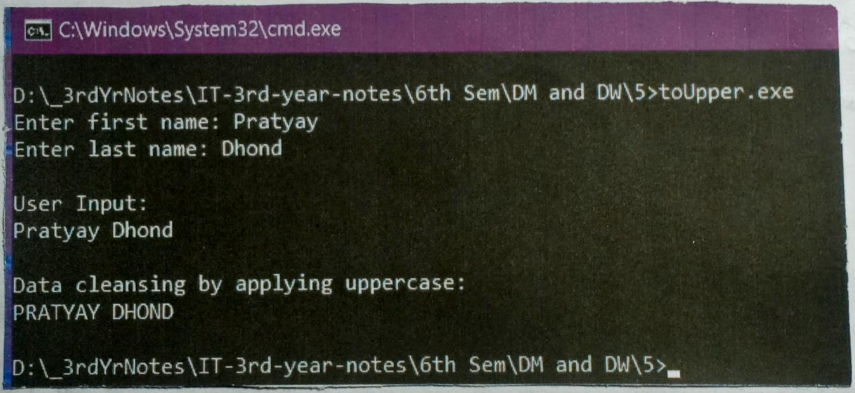
SYNTAX:

LOWER() → converts to lower case

mary jane

UPPER → Converts to uppercase

MARY JANE



```
D:\_3rdYrNotes\IT-3rd-year-notes\6th Sem\DM and DW\5>toUpper.exe
Enter first name: Pratyay
Enter last name: Dhond

User Input:
Pratyay Dhond

Data cleansing by applying uppercase:
PRATYAY DHOND

D:\_3rdYrNotes\IT-3rd-year-notes\6th Sem\DM and DW\5>
```

Output : toUpper.exe

PROPER() → Converts all text to proper ~~text~~ case.

Mary Jane.

PROGRAM: To convert input string into an uppercase string.

Logic: Looping through the characters of the input string, and checking whether the character lies in the ASCII range 97 - 122 (all lower case lies in this range) If the character is found to be in this range then program converts it to an uppercase by subtracting 32 from ASCII values.

CODE:

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

string upper(string str) {
    for(int i=0; i< str.length(); i++){
        int ch= str[i];
        if(ch>96 && ch<123){
            str[i]  =ch-32;
        }
    }
    return str;
}

int main(){
    string firstName, lastName;
    cout<<"Enter First Name: ";
    cin>> firstName ;
```

Conclusion: Thus, I implemented data cleansing by applying uppercase of firstName and lastName in C++.

```
    cout<<"Enter last name: ";
    cin >> lastName;

    cout<<"User Input: \n";
    cout << firstName << " " << lastName << endl;
    firstName ~~E~~ = upper(FirstName);
    lastName ~~E~~ = upper(lastName);
    cout<<"\nData cleansing by applying uppercase: \n";
    cout<< firstName << " " << lastName;

    return 0;
}
```

Conclusion: Thus, I implemented data cleansing by applying uppercase of firstName and last name in C++.