STRING Tutorial

S.D

①

Q. Write a program to find the length of string.

```
#include<iostream.h>

int main( )
{
    char str[80];

    cout<<"Enter string: ";
    cin.getline(str, 80);

    int i; //Hold length of string

    for(i = 0; str[i] != '\0'; i++);

    cout << "Length of string is: " << i;

    return 0;
}
```

Q. Write a program to display string from backward.

```
#include<iostream.h>
int main( )
{
    char str[80];

    cout<<"Enter string: ";
    cin.getline(str, 80);

    int l; //Hold length of string

    //Find the length of the string
    for(l = 0; str[l] != '\0'; l++);

    //Display the string backwards
    for(int i = l - 1; i >= 0; i--)
    {
        cout << str[i];
    }

    return 0;
}
```

STRING Tutorial

Q. Write a program to count number of words in string.

```
#include<iostream>
int main( )
{
        char str[80];

        cout << "Enter a string: ";
        cin.getline(str,80);

        int words = 0; // Holds number of words

        for(int i = 0; str[i] != '\0'; i++)
        {
                if (str[i] == ' ') //Checking for spaces
                {
                        words++;
                }
        }

        cout << "The number of words = " << words+1 << endl;

        return 0;
}
```

Q. Write a program to concatenate one string contents to another.

```
#include<iostream>
int main( )
{
     char str1[80], str2[80];

     cout<<"Enter first string: ";
     cin.getline(str1, 80);

     cout<<"Enter second string: ";
     cin.getline(str2, 80);

     int l = 0; //Hold length of first string

     //Find length of first string.
     for(l = 0; str1[l] != '\0'; l++);

     //Adding second string content in first
     for(int i = 0; str2[i] != '\0'; i++)
     {
          str1[l++] = str2[i];
```

```
    }

    str1[l] = '\0';

    cout << "\nThe first string after adding second string
    content:\n\n" << str1;

    return 0;
}
```

Q. Write a C++ program to compare two strings they are exact equal or not.

```
#include<iostream>
int main( )
{
    char str1[80], str2[80];

    cout<<"Enter first string: ";
    gets(str1);

    cout<<"Enter second string: ";
    gets(str2);

    int i;
    for (i = 0; str1[i] == str2[i] && str1[i]!= '\0' && str2[i]
!= '\0'; i++);

    if(str1[i] - str2[i] == 0)
        cout << "Strings are equal";
    else
        cout << "Strings are not equal";

    return 0;
}
```

Q. Write a program to check a string is palindrome or not.

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

int main( )
{
    char str[80];

    cout<<"Enter string: ";
    cin.getline(str, 80);

    int l; //Hold length of string

    //finding length of string
    for(l = 0; str[l] != '\0'; l++);

    //Comparing first element with last element till middle
    of string
    int i;
    for(i = 0; (i < l/2) && (str[i] == str[l - i - 1]);i++)

    if(i == l/2)
        cout << "Palindrome";
    else
        cout << "Not a palindrome";

    return 0;
}
```

Q. Write a program to find a substring within a string. If found display its starting position.

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

int main( )
{
    char str1[80], str2[80];

    cout<<"Enter first string: ";
    cin.getline(str1, 80);

    cout<<"Enter second string: ";
    cin.getline(str2, 80);

    int l = 0; //Hold length of second string

    //finding length of second string
    for(l = 0; str2[l] != '\0'; l++);

    int i, j;

    for(i = 0, j = 0; str1[i] != '\0' && str2[j] != '\0';i++)
    {
        if(str1[i] == str2[j])
        {
            j++;
            }
            else
            {
                j = 0;
            }
        }

        if(j == l)
            cout<<"Substring found at position "<< i - j + 1;
        else
            cout<<"Substring not found";

        return 0;
    }
```

Q. Write a C++ program to reverse a string.

```
    #include<iostream>
    int main( )
    {
        char str[80];

        cout<<"Enter string: ";
        cin.getline(str, 80);

        int l; //Hold length of string
        for(l = 0; str[l] != '\0'; l++); //finding length of string

        int temp;
        for(int i = 0, j = l - 1; i < l/2; i++, j--)
        {
            temp = str[i];
            str[i] = str[j];
            str[j] = temp;
        }

        cout << "Reverse string: " << str << endl;

        return 0;
    }
```

Note-5
SD

DATA FILE HANDLING IN C++

File

- A file is a stream of bytes stored on some secondary storage devices.
- **Text file:** A text file stores information in readable and printable form. Each line of text is terminated with an **EOL** (End of Line) character.
- **Binary file:** A binary file contains information in the non-readable form i.e. in the same format in which it is held in memory.

File Stream

- **Stream:** A stream is a general term used to name flow of data. Different streams are used to represent different kinds of data flow.
- There are three file I/O classes used for file read / write operations.
  - **ifstream** - can be used for read operations.
  - **ofstream** - can be used for write operations.
  - **fstream** - can be used for both read & write operations.
- **fstream.h**:
- This header file includes the definitions for the stream classes ifstream, ofstream and fstream. In C++ **file input output** facilities implemented through fstream.h header file.
- It contain predefines set of operation for handling file related input and output, fstream class ties a file to the program for input and output operation.
- A file can be opened using:
  - **By the constructor method**. This will use default streams for file input or output. This method is preferred when file is opened in input or output mode only. Example : **ofstream file("student.dat"); or ifstream file("student.dat");**
  - **By the open() member function** of the stream. It will preferred when file is opened in various modes i.e ios::in, ios::out, ios::app, ios::ate etc.
    e.g **fstream file;**
    file.open("book.dat", ios::in | ios::out | ios::binary);

File modes:

- **ios::out** It open file in output mode (i.e write mode) and place the file pointer in beginning, if file already exist it will overwrite the file.
- **ios::in** It open file in input mode(read mode) and permit reading from the file.
- **ios::app** It open the file in write mode, and place file pointer at the end of file i.e to add new contents and retains previous contents. If file does not exist it will create a new file.
- **ios::ate** It open the file in write or read mode, and place file pointer at the end of file i.e input/ output operations can performed anywhere in the file.
- **ios::trunc** It truncates the existing file (empties the file).
- **ios::nocreate**If file does not exist this file mode ensures that no file is created and open() fails.
- **ios::noreplace**If file does not exist, a new file gets created but if the file already exists, the open() fails.
- **ios::binary**Opens a file in binary mode.

eof(): This function determines the end-of-file by returning true(non-zero) for end of file otherwise returning false(zero).

close(): This function terminates the connection between the file and stream associated with it.

Stream_object.close(); **e.g file.close();**

Text File functions:

Char I/O :

- **get()** – read a single character from text file and store in a buffer. e.g **file.get(ch);**
- **put()** - writing a single character in textfile e.g. **file.put(ch);**
- **getline()** - read a line of text from text file store in a buffer. e.g **file.getline(s,80);**

- We can also use **file>>ch** for reading and **file<<ch** writing in text file. But >> operator does not accept white spaces.

Binary file functions:

- **read()**- read a block of binary data or reads a fixed number of bytes from the specified stream and store in a buffer.

  Syntax : Stream_object.read((char *)& Object, sizeof(Object)); e.g file.read((char *)&s, sizeof(s));

- **write()** – write a block of binary data or writes fixed number of bytes from a specific memory location to the specified stream.

  Syntax : Stream_object.write((char *)& Object, sizeof(Object)); e.g file.write((char *)&s, sizeof(s));

Note:

Both functions take two arguments.

- The first is the address of variable, and the second is the length of that variable in bytes. The address of variable must be type cast to type char*(pointer to character type)
- The data written to a file using write( ) can only be read accurately using read( ).

File Pointer: The file pointer indicates the position in the file at which the next input/output is to occur.

Moving the file pointer in a file for various operations viz modification, deletion , searching etc. Following functions are used:

seekg(): It places the file pointer to the specified position in input mode of file.
e.g file.seekg(p,ios::beg); or file.seekg(-p,ios::end), or file.seekg(p,ios::cur) i.e to move to **p** byte position from beginning, end or current position.

seekp(): It places the file pointer to the specified position in output mode of file.
e.g file.seekp(p,ios::beg); or file.seekp(-p,ios::end), or file.seekp(p,ios::cur) i.e to move to **p** byte position from beginning, end or current position.

tellg(): This function returns the current working position of the file pointer in the input mode. **e.g int p=file.tellg();**

tellp(): This function returns the current working position of the file pointer in the output mode.
e.f int p=file.tellp();

Steps To Create A File

- Declare an object of the desired file stream class(ifstream, ofstream, or fstream)
- Open the required file to be processed using constructor or open function.
- Process the file.
- Close the file stream using the object of file stream.

General program structure used for creating a Text File

To create a text file using strings I/O

```
#include<fstream.h> //header file for file
operations
void main()
{
char s[80], ch;
ofstream file("myfile.txt"); //open myfile.txt in default output mode
do
{
cout<<"\n enter line of text";
gets(s); //standard input
file<<s;  // write in a file myfile.txt
cout<<"\n more input y/n";
cin>>ch;
}while(ch!='n'||ch!='N');
file.close();
} //end of main
```

To create a text file using characters I/O

```
#include<fstream.h>   //header file for file operations
void main()
{
char  ch;
ofstream file("myfile.txt"); //open myfile.txt in default output mode
do{
ch=getche();
if (ch==13)  //check if character is enter key
cout<<'\n';
else
file<<ch;  // write a character in text file 'myfile.txt '
} while(ch!=27); // check for
escape key file.close();
} //end of main
```

Text files in input mode:

To read content of 'myfile.txt' and display it on monitor.

```
#include<fstream.h>   //header file for file operations
void main()
{
char  ch;
ifstream file("myfile.txt"); //open myfile.txt in default input mode
while(file)
{
file.get(ch) // read a character from text file
'myfile.txt' cout<<ch; // write a character in text
file 'myfile.txt '
}
file.close();
} //end of main
```

Write function definition for DISP3CHAR() in C++ to read the content of a text file KIDINME.TXT, and display all those words, which have three characters in it.

```
DISP3CHAR()
  {   ifstream Fil; Fil.open("KIDINME.TXT");
      char W[20]; Fil>>W;
     while(!Fil.eof()) // OR while(Fil)
     {   if (strlen(W)) == 3)
                 cout<<W<< " "; Fil>>W;
     }
     Fil.close(); //Ignore
  }
```

Write a user defined function word_count() in C++ to count how many words are present in a text file named "opinion.txt".

```
void word_count()
{ ifstream i;char ch[20];int c=0; i.open("opinion.txt ");
while(!i.eof())
{
i>>ch; c=c+1;
}
cout<<" Total number of words present in the text file are: "<<c;
}
```

Write function definition for TOWER() in C++ to read the content of a text file WRITEUP.TXT, count the presence of word TOWER and display the number of occurrences of this word.

```
void TOWER()
{
int count=0;
ifstream f("WRITEUP.TXT"); char s[20];
while (!f.eof())
{
     f>>s;
     if (strcmpi(s,"TOWER")==0) count++;
}
cout<<count; f.close();
}
```

Write the function AECount( ) in C++, which should read character of a text file NOTES.txt, should count and display the occurrence of alphabets A and E (including small case a and e too)

```
void AECount( )
{
 char Ch;
 ifstream fcin("NOTES.txt"); int count1=0,count2=0; while(!fcin.eof( ))
  {
                        fcin.get(CH);
```

```
        if(Ch= ='A'||Ch=='a')
            count1++;
        else if (Ch= ='E'||Ch=='e')
            count2++;
    }
                        fcin.close( );
    cout<<"A: "<<count1<<endl; cout<<"E:"<<count2<<endl;
}
```

Write a function in C++ to count the no. of "He" or "She" words present in a text file "STORY. TXT".

```
void COUNT ( )
{
  ifstream Fil ("STORY.TXT"); char STR [10];
  int count = 0; while (!Fil.eof ( ))
  {    Fil>>STR ;
     if (strcmp (STR, "He") ==0 | | strcmp (STR, "She") = =0) count++;
   }
   cout<<"Count of He/She in file : "<<count<<end1; Fil.close( ); //Ignore
}
```

Write a function in C++ to count the words "this" and "these" present in a text file "ARTICLE.TXT".

```
void COUNT ( )
{
ifstream Fil; // ifstreamFil("ARTICLE.TXT");
Fil. open("ARTICLE.TXT");
char Word[80] ,Ch;
int Cl =0, C2 = 0, I=O;
while(Fil.get(Ch))
{ if (Ch! = ' ')
    Word[I++] = Ch; else
                    { Word[I] = '\0';
   if (strcmp (Word,"this")==0)
                        Cl++;
   else if (strcmp(Word,"these")==0)
      C2++; I=0;
 }
}
cout<<"Count of -this- in file:"<<Cl;
cout<<"Count of -these- in file:"<<C2;
// OR cout<<"Count of -this- and –these- in file: "<<Cl+C2;
Fil.close( );
}
```

Write a function in C++ to count the words "to" and "the" present in a text file "POEM.TXT".

```
void COUNT( )
{
ifstream Fil;
Fil. open ("POEM.TXT");
                        //OR ifstream Fill("POEM.TXT");
char Word[8O], Ch; int Cl =0, C2=0, i=0;
while(Fil.get(Ch))
{
if (Ch! = ' ') Word[i++] = Ch; else
{ Word[i] = '\0';
 if (strcmp (Word, "to") ==0)
   Cl++;
```

```
                    else if (strcmp (Word, "the") ==0)
     C2++; i=0;
  }
 }
 cout<<"Count of -to- in file:"<<Cl; cout<<"Count of -the- in file:"<<C2;
     //OR cout«"Count of -to- and -the- in Fil.close( );
 }
```

Write a function to count the number of words present in a text file named "PARA.TXT". Assume that each word is separated by a single blank/space character and no blanks/spaces in the beginning and end of the file.

```
void WordsCount( )
{ clrscr( );
  ifstream fin("PARA.TXT",ios::in); char ch;
  int Words=1; if(!fin)
  { cout<<"No words at all in the file";
   exit(0);
  }
   while(fin)
  { fin.get(ch);
   if(ch= =' ')
          Words++;
  }
  cout<<"\nTotal number of Words in the file = "<<Words; getch( );
}
```

Write a function in C++ to count the number of alphabets present in a text file "NOTES.TXT".

```
void CountAlphabet()
{
ifstream FIL("NOTES.TXT");
int CALPHA=0;
char CH=FIL.get( ); while (!FIL.eof( ))
{   if(isalpha(CH)) CALPHA++;
                              CH=FIL.get( );
}
cout<<"No. of Alphabets:"<<CALPHA <<endl;
FIL.close( );
}
```