

Computer Programming

Dr. Deepak B Phatak
Dr. Supratik Chakraborty
Department of Computer Science and Engineering
IIT Bombay

Session: Handling Data in Text Files

Quick Recap


- C++ program can handle text data
- We studied the functions `printf()` and `scanf()`
 - `sprintf()`, `sscanf()`; `fscanf()` , `fprintf()`
- Earlier, we had seen how files are handled in C++
 - File pointers, functions like `fopen()` `fclose()`, `feof()`, ...

Overview



- In this session, we will use our learning to handle data from in text files

Example of a spread-sheet

<div>  <div>Nimbus Sans L</div> <div>10</div> <div>A A A</div> <div></div> <div></div> </div>					
<div> <div>A10</div> <div>f(x) ✖ ✔</div> <div>10115</div> </div>					
	A	B	C	D	E
1	10101	Anil Shah	112	12.5	
2	10102	Amit Jadhav	111	15	
3	10103	Shephali Pandya	112	17	
4	10104	Rajesh Mashruwala	111	19	
5	10105	Nandan Meshram	111	16	
6	10106	Avinash Adsule	112	14	
7	10107	Srikant Rao	112	14.5	
8	10108	nilmani Raut	111	11.5	
9	10110	Rajesh Singh	112	10	
10	10115	Ketan Maheshwari	111	12	
11					
12					

CSV data (Comma Separated Values)



10101,Anil Shah,112,12.5
10102,Amit Jadhav,111,15
10103,Shefali Pandya,112,17
10104,Rajesh Mashruwala,111,19
10105,Nandan Meshram,111,16
10106,Avinash Adsule,112,14
10107,Srikant Rao,112,14.5
10108,Nilmani Raut,111,11.5
10110,Rajesh Singh,112,10
10115,Ketan Maheshwari,111,12

A program to process data from a CSV file



- Read one line from the input file in a string
 - `char linestr[80];`
For example, the first line is: 10101,Anil Shah,112,12.5
- Separate the four parts in four different strings
- Convert each part in internal form in appropriate variables
`int sr; char sn[30]; int sb; float sm;`
- Now put these four values, separated by blank spaces, together in a string outstr (`char outstr[80];`)
- write this string to the output file
- Repeat this procedure to process all lines from the input file

Program logic


```
Read one line of input file in linestr
While (not end-of-file for the input file){
    Process input string, separate parts in four strings
    convert each part and store in an appropriate variable
    Prepare an output text string with these four values
    (values should be separated by blank spaces)
    write this string ( outstr) to output file
    read next input line in linestr
}
```

Program logic

Read one line of input file in `linestr`

While (not end-of-file for the input file){

Process input string, separate parts in four strings
convert each part and store in an appropriate variable
Prepare an output text string with these four values
(values should be separated by blank spaces)
write this string (`outstr`) to output file
read next input line in `linestr`



}

Separating parts in four strings

|1|0|1|0|1|,|A|n|i|l| |S|h|a|h|,|1|1|2|,|1|2|.|5|



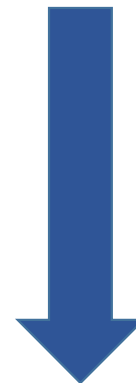
scroll



sname



sbatch



smarks

Program ...



```
#include <iostream>
#include <cstring>
#include <cstdio>
using namespace std;
int main() {
    char linestr[80]; char outstr[80];
    char sroll[6], sname[30], sbatch[4], smarks[10];
    int sr; char sn[30]; int sb; float sm;
    int i,j,k,N =0;
```

Program ...

```
FILE *fpin; FILE* fpout;
fpin = fopen("CSV_data.txt", "r" );
if (fpin == NULL){
    cout << "Could not open file" << endl;
    return -1;
}
fpout = fopen ("marks_data.txt", "w");
if (fpout == NULL){
    cout << "Could not create output file" << endl;
    return -1;
}
```

Program ...



```
/*Input file is open at this point, read lines one by one */  
fgets(linestr, 79, fpin);  
while (!feof (fpin)){  
    /* valid string, separate the parts */  
    i =0; k =0;
```

Program ...

```
while ((sroll[i++] = linestr[k++]) != ',');  
sroll[i-1]='\0'; i=0;  
while ((sname[i++] = linestr[k++])!= ',');  
for (j = i-1; j<29; j++) sname[j] = ' '  
sname[29]='\0'; i=0;  
while ((sbatch[i++] = linestr[k++]) != ',');  
sbatch[i-1] = '\0'; i=0;  
while((smarks[i++]=linestr[k++]) != '\0');
```

Program ...



```
/*extract relevent values from these strings */  
sscanf (sroll, "%d", &sr);  
sscanf(sname,"%s", sn);  
sscanf(sbatch,"%d", &sb);  
sscanf(smarks, "%f", &sm);
```

Program ...



```
    sprintf(outstr, "%5d %30s %3d %5.2f\n",sr,sn,sb,sm);  
    fputs(outstr, fpout);  
    printf("%s", ostr);  
    fgets(linestr, 79, fpin);  
    N=N+1;  
} //End of while loop, Entire input file has been processed
```

Program ...

```
cout << "\ninput file has been read and printed\n";  
cout << "output file Marks_data.txt created\n";  
cout << N << " records written to output file\n";  
fclose(fpin); fclose(fpout);  
return 0;  
}
```


Code Blocks execution results

```
M:\codeblocks\processing_CSV_data\bin\Debug\processing_CSV_data.exe
10101          Anil 112 12.50
10102          Amit 111 15.00
10103      Shefali 112 17.00
10104          Rajesh 111 19.00
10105          Nandan 111 16.00
10106      Avinash 112 14.00
10107      Srikant 112 14.50
10108      Nilmani 111 11.50
10110          Rajesh 112 10.00
10115          Ketan 111 12.00

input file has been read and printed
output file Marks_data.txt created
10 records written to output file

Process returned 0 (0x0)    execution time : 0.089 s
Press any key to continue.
```

Summary



- We wrote a program to process data in CSV format from a file
- In the next two sessions:
 - We will create a binary file to store the database of students
 - We will process the records in this file by using 'direct'access'
- Refer to C++ tutorials and reference section on the web at:
<http://www.cplusplus.com/reference/cstdio>
- Read about all the relevant file functions