

# Streams and File I/O

Computer Programming for Engineers

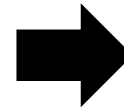
Week14

Nov/30/2017

# Problem #1 A Neat File

- Read numbers from the given file, and arrange the numbers adapting the given format
  - File name: rawdata.txt
- Save the numbers to a file
  - File name: neat.txt
- Format
  - Print sign of the number (ex: +12, -3.1)
  - Print five numbers after decimal point (ex: +3.00000)
  - Right align numbers with 12 widths (ex: \_\_\_\_-4.23000)

```
3.512 -54_+123 0.5123↵
4_124_____4214.3_-123.43_____53.000↵
12.3__+25__-46.53_____12.534↵
```



```
____+3.51200↵
____-54.00000↵
__+123.00000↵
____+0.51230↵
____+4.00000↵
____+124.00000↵
__+4214.30000↵
__-123.43000↵
____+53.00000↵
____+12.30000↵
____+25.00000↵
____-46.53000↵
____+12.53400↵
```

# Problem #2 Excel Printer

- Create a well arranged table
- First, get table size
  - #N #N
- Get cell information and string data from input
  - #N #N STRING
- Create a table with the cells and save it to a file
  - File name: test.txt

# Problem #2 Excel Printer

- Each row is separated with '-'
  - A row is with four digits, 1~9999
- Each column is separated with '|'
  - A column is with two alphabets, A~Z~AA~ZZ
- An item of each cell is left aligned
- Table is two-dimension array of cells
  - Create cell arrays according to the table size
  - Create cell item dynamically
  - Be careful about memory management (delete item)

# Problem #2 Excel Printer

	A	B	C	D	E
1	jirej				
2		ere	gfewegewege		
3					
4			12215252		
5					

# Problem #2 Excel Printer

	A	B	C	D	E
1	jirej				
2		ere	gfewegewege		
3					
4			12215252		
5					

Class Table

# Problem #2 Excel Printer

	A	B	C	D	E
1	jirej				
2		ere			
3					
4			12215252		
5					

# Problem #2 Excel Printer

	A	B	C	D	E
1	jirej				
2		ere	gfewegewege		
3					
4			12215252		
5					

Four texts



# Problem #2 Excel Printer

	A	B	C	D	E
1	jirej				
2		ere	gfewegewege		
3					
4			12215252		
5					

Left aligned

# Problem #2 Excel Printer

	A	B	C	D	E
1	jirej				
2		ere	gfewegewege		
3					
4			12215252		
5					

Min two texts

Thank You

```
1 #include <iostream>
2 #include <fstream>
3 #include <cstdlib>
4 #include <iomanip>
5
6 using namespace std;
7
8 void makeNeat(ifstream& messyFile, ofstream& neatFile,
9     int numberAfterDecimalpoint, int fieldWidth)
10 {
11     neatFile.setf(ios::fixed);
12     neatFile.setf(ios::showpoint);
13     neatFile.setf(ios::showpos);
14     neatFile.precision(numberAfterDecimalpoint);
15
16     cout.setf(ios::fixed);
17     cout.setf(ios::showpoint);
18     cout.setf(ios::showpos);
19     cout.precision(numberAfterDecimalpoint);
20
21     double next;
22     while (messyFile >> next)
23     {
24         neatFile << setw(fieldWidth) << next << endl;
25     }
26 }
27
28 int main()
29 {
30     string data;
31     ifstream fin;
32     ofstream fout;
33
34     fout.open("rawdata.txt");
35     while (1) {
36         getline(cin, data);
37         if (data.compare("exit") == 0)
38             break;
39         fout << data << endl;
40     }
41     fout.close();
42
43     fin.open("rawdata.txt");
44     fout.open("neat.txt");
45
46     makeNeat(fin, fout, 5, 12);
47
48     fin.close();
49     fout.close();
50
51     fin.open("neat.txt");
52     string tmp;
53     while (getline(fin, tmp))
54         cout << tmp << endl;
55     return 0;
56 }
```

```
1 #include <iostream>
2 #include <fstream>
3 #include <string>
4 #include <sstream>
5
6 using namespace std;
7
8 class Cell;
9 class Table;
10
11 class Cell
12 {
13     string data;
14 public:
15     Cell(string data);
16     string get_data();
17 };
18
19 Cell::Cell(string data) : data(data) { }
20
21 string Cell::get_data() {
22     return data;
23 }
24
25 class Table
26 {
27     int max_row_size, max_col_size;
28     Cell*** data_table;
29
30     string repeat_char(int n, char c);
31     string col_num_to_str(int n);
32 public:
33     Table(int max_row_size, int max_col_size);
34     ~Table();
35
36     void reg_cell(Cell* c, int row, int col);
37
38     string get_data(int row, int col);
39
40     string print_table();
41 };
42
43 Table::Table(int max_row_size, int max_col_size) :
44     max_row_size(max_row_size), max_col_size(max_col_size)
45 {
46     data_table = new Cell**[max_row_size];
47     for (int i = 0; i < max_row_size; i++) {
48         data_table[i] = new Cell*[max_col_size];
49         for (int j = 0; j < max_col_size; j++) {
50             data_table[i][j] = NULL;
51         }
52     }
53 }
54
55 Table::~Table()
56 {
```

```
57     for (int i = 0; i < max_row_size; i++) {
58         for (int j = 0; j < max_col_size; j++) {
59             if (data_table[i][j]) delete data_table[i][j];
60         }
61     }
62     for (int i = 0; i < max_row_size; i++) {
63         delete[] data_table[i];
64     }
65     delete[] data_table;
66 }
67
68 void Table::reg_cell(Cell* c, int row, int col) {
69     if (!(row < max_row_size && col < max_col_size)) return;
70
71     if (data_table[row][col]) {
72         delete data_table[row][col];
73     }
74     data_table[row][col] = c;
75 }
76
77 string Table::get_data(int row, int col) {
78     if (row < max_row_size && col < max_col_size && data_table[row][col]) {
79         return data_table[row][col]->get_data();
80     }
81     return "";
82 }
83
84 ostream& operator<< (ostream& o, Table& table) {
85     o << table.print_table();
86     return o;
87 }
88
89 string Table::print_table()
90 {
91     string total_table;
92
93     int* col_max_wide = new int[max_col_size];
94     for (int i = 0; i < max_col_size; i++) {
95         int max_wide = 2;
96         for (int j = 0; j < max_row_size; j++) {
97             if (data_table[j][i] && data_table[j][i]->get_data().length() >
98                 max_wide) {
99                 max_wide = data_table[j][i]->get_data().length();
100             }
101             col_max_wide[i] = max_wide;
102         }
103
104         total_table += "    ";
105         int total_wide = 4;
106         for (int i = 0; i < max_col_size; i++) {
107             if (col_max_wide[i]) {
108                 int max_len = max(2, col_max_wide[i]);
109                 total_table += "|" + col_num_to_str(i);
110                 total_table += repeat_char(max_len - col_num_to_str(i).length(), ' ');
111             }
112         }
113     }
114 }
```

```
111
112     total_wide += (max_len + 1);
113 }
114 }
115
116 total_table += "\n";
117
118 for (int i = 0; i < max_row_size; i++) {
119     total_table += repeat_char(total_wide, '-');
120     total_table += "\n" + to_string(i + 1);
121     total_table += repeat_char(4 - to_string(i + 1).length(), ' ');
122
123     for (int j = 0; j < max_col_size; j++) {
124         if (col_max_wide[j]) {
125             int max_len = max(2, col_max_wide[j]);
126
127             string s = "";
128             if (data_table[i][j]) {
129                 s = data_table[i][j]->get_data();
130             }
131             total_table += "|" + s;
132             total_table += repeat_char(max_len - s.length(), ' ');
133         }
134     }
135     total_table += "\n";
136 }
137
138 return total_table;
139 }
140
141 string Table::repeat_char(int n, char c)
142 {
143     string s = "";
144     for (int i = 0; i < n; i++)
145         s.push_back(c);
146
147     return s;
148 }
149
150 string Table::col_num_to_str(int n)
151 {
152     string s = "";
153     if (n < 26) { s.push_back('A' + n); }
154     else {
155         char first = 'A' + n / 26 - 1;
156         char second = 'A' + n % 26;
157
158         s.push_back(first);
159         s.push_back(second);
160     }
161
162     return s;
163 }
164
165 int main()
166 {
```

```
167     int X, Y;
168
169     cin >> X >> Y;
170     cin.ignore(100, '\n');
171
172     Table table(X, Y);
173
174     while (true) {
175         string tmp, data;
176         stringstream sstream;
177         int x, y;
178
179         getline(cin, tmp);
180
181         sstream.str(tmp);
182         sstream >> x;
183
184         if (x == -1)
185             break;
186
187         sstream >> y >> data;
188
189         table.reg_cell(new Cell(data), x, y);
190     }
191
192     ofstream out("test.txt");
193     out << table;
194     out.close();
195
196     ifstream in("test.txt");
197     string tmp;
198     while (getline(in, tmp))
199         cout << tmp << endl;
200 }
```