

```
Script started on 2023-07-27 12:23:54-05:00 [TERM="xterm" TTY="/dev/pts/2" COLUMNS=
bj94684@ares:~$ pwd
/home/students/bj94684
bj94684@ares:~$ cat box.info
/*****
```

```
*
* NAME:   Jose Barron                CLASS:  CSC122-002      *
*
* Lab: A box upon ye                Level:   5              *
* Option:                          Level:  + 0             *
*
* Total Level:  5.0                *
*
* This program is designed to create a frame around a phrase given by a *
* user, either from keyboard input or a file. The user has the ability *
* to choose how the words are centered (center, left, or right) and what *
* symbol the frame is made of. The words are processed with a vector of *
* strings which are the words then with the use of the FramedWord class *
* the words are framed.
*
*****/
```

```
bj94684@ares:~$ show-code strextra.h
```

strextra.h:

```
1  #ifndef STREXTRA_H_INC
2  #define STREXTRA_H_INC
3
4  #include<iostream>
5  #include<string>
6  #include<vector>
7  #include<cctype>
8
9  using namespace std;
10
11
12 inline std::string tolower_str(std::string s)
13 {
14     std::string t = s;
15     for(std::string::size_type pos = 0; pos != s.length(); ++pos)
16     {
17         t[pos] = static_cast<char>( tolower(t[pos]) );
18     }
19     return t;
20 }
21
22
23 inline std::vector<std::string::size_type> find(std::string s, char t,
24                                                bool case_sensitive = true)
25 {
26     if( ! case_sensitive)
27     {
28         s = tolower_str(s);
```

```
29         t = static_cast<char>( tolower(t) );
30     }
31
32     std::vector<string::size_type> num;
33     for (std::string::size_type pos = 0; pos != s.length(); ++pos)
34     {
35         if (static_cast<char>( s[pos] ) == t )
36         {
37             num.push_back(pos);
38         }
39     }
40     return num;
41 }
42
43 inline bool str_is_incl(std::string s, std::string t, std::string::size_ty
44                       std::string::size_type end)
45 {
46     std::string nr = s.substr(beg, end-beg+1);
47     bool tf;
48     if ( nr == t)
49     {
50         return tf = true;
51     }
52     return tf = false;
53 }
54
55 inline std::string::size_type strcom(std::vector<string::size_type> b,
56                                     std::vector<string::size_type> e,
57                                     std::string s, std::string t)
58 {
59     string::size_type num;
60     for(auto p : b)
61     {
62         for(auto p1 : e)
63         {
64             if ( p < p1)    // The starter position can never be greater
65             {
66                 bool ys = str_is_incl(s, t, p, p1);
67                 if ( ys)
68                 {
69                     return num = p;
70                 }
71             }
72         }
73     }
74     return num = s.length();
75 }
76
77 std::string::size_type find(std::string s, std::string t, bool c_s = true)
78
79
80
81 #endif
```

```
bj94684@ares:~$ show-code strextra.h
```

strextra.h:

```
1  #ifndef STREXTRA_H_INC
2  #define STREXTRA_H_INC
3
4  #include<iostream>
5  #include<string>
6  #include<vector>
7  #include<cctype>
8
9  using namespace std;
10
11
12 inline std::string tolower_str(std::string s)
13 {
14     std::string t = s;
15     for(std::string::size_type pos = 0; pos != s.length(); ++pos)
16     {
17         t[pos] = static_cast<char>( tolower(t[pos]) );
18     }
19     return t;
20 }
21
22
23 inline std::vector<std::string::size_type> find(std::string s, char t,
24                                                bool case_sensitive = true)
25 {
26     if( ! case_sensitive)
27     {
28         s = tolower_str(s);
29         t = static_cast<char>( tolower(t) );
30     }
31
32     std::vector<string::size_type> num;
33     for (std::string::size_type pos = 0; pos != s.length(); ++pos)
34     {
35         if (static_cast<char>( s[pos] ) == t )
36         {
37             num.push_back(pos);
38         }
39     }
40     return num;
41 }
42
43 inline bool str_is_incl(std::string s, std::string t, std::string::size_ty
44                        std::string::size_type end)
45 {
46     std::string nr = s.substr(beg, end-beg+1);
47     bool tf;
48     if ( nr == t)
49     {
```

```
50         return tf = true;
51     }
52     return tf = false;
53 }
54
55 inline std::string::size_type strcom(std::vector<string::size_type> b,
56                                     std::vector<string::size_type> e,
57                                     std::string s, std::string t)
58 {
59     string::size_type num;
60     for(auto p : b)
61     {
62         for(auto p1 : e)
63         {
64             if ( p < p1) // The starter position can never be greater
65             {
66                 bool ys = str_is_incl(s, t, p, p1);
67                 if ( ys)
68                 {
69                     return num = p;
70                 }
71             }
72         }
73     }
74     return num = s.length();
75 }
76
77 std::string::size_type find(std::string s, std::string t, bool c_s = true)
78
79
80
81 #endif
```

bj94684@ares:~\$ show-code box.cpp

box.cpp:

```
1  #include <iostream>
2  #include <fstream>
3  #include <vector>
4  #include <cctype>
5  #include <limits>
6  #include "strextra.h"
7
8  using namespace std;
9
10 inline string::size_type find_space( string s, char t)
11 {
12     string::size_type num = s.length();
13     for (string::size_type pos = 0; pos < s.length(); ++pos)
14     {
15         if (static_cast<char>( s[pos] ) == t )
16         {
```

```

17         return num = pos;
18     }
19 }
20 return num;
21 }
22
23 vector<string> word_get(string sen);
24
25 class FrameWord
26 {
27 private:
28     vector<string> words;
29     short max_length_word;
30     vector<string> framed_words;
31     vector<string> framed_words_left;
32     vector<string> framed_words_right;
33     char frame_c;
34
35     void center_word(string & word) const;
36     void center_word_left(string & word) const;
37     void center_word_right(string & word) const;
38
39 public:
40     FrameWord(const vector<string> & inputted_words, char f_c);
41     void frame_words();
42     void frame_words_left();
43     void frame_words_right();
44     void display_frame() const;
45     void display_frame_left() const;
46     void display_frame_right() const;
47
48 };
49
50 FrameWord::FrameWord(const vector<string> & inputted_words, char f_c)
51     : words(inputted_words), max_length_word(0), frame_c(f_c)
52 {
53     for (const auto & word : words)
54     {
55         max_length_word = max(max_length_word, static_cast<short>(word.length()));
56     }
57
58 }
59
60 void FrameWord::center_word(string & word) const
61 {
62     short space = max_length_word - static_cast<short>(word.length());
63     short left_side = space / 2;
64     short right_side = space - left_side;
65     word = frame_c + string(left_side + 1, ' ') + word + string(right_side
66 }
67
68 void FrameWord::center_word_left(string & word) const
69 {
70

```

```

71     short space = max_length_word - static_cast<short>(word.length());
72     word = frame_c + word + string(space + 2, ' ') + frame_c;
73 }
74
75 void FrameWord::center_word_right(string & word) const
76 {
77     short space = max_length_word - static_cast<short>(word.length());
78     word = frame_c + string(space + 1, ' ') + word + " " + frame_c;
79 }
80
81 void FrameWord::frame_words()
82 {
83     for (const auto & word : words)
84     {
85         string framed_word = word;
86         center_word(framed_word);
87         framed_words.push_back(framed_word);
88     }
89 }
90
91 void FrameWord::frame_words_left()
92 {
93     for (const auto & word : words)
94     {
95         string framed_word_left = word;
96         center_word_left(framed_word_left);
97         framed_words_left.push_back(framed_word_left);
98     }
99 }
100
101 void FrameWord::frame_words_right()
102 {
103     for (const auto & word : words)
104     {
105         string framed_word_right = word;
106         center_word_right(framed_word_right);
107         framed_words_right.push_back(framed_word_right);
108     }
109 }
110
111 void FrameWord::display_frame() const
112 {
113     short frame_size = max_length_word + 4;
114     cout << string(frame_size, frame_c) << '\n';
115     for (const auto & word : framed_words)
116     {
117         cout << word << endl;
118     }
119     cout << string(frame_size, frame_c) << '\n';
120 }
121
122 void FrameWord::display_frame_left() const
123 {
124

```

```

125     short frame_size = max_length_word + 4;
126     cout << string(frame_size, frame_c) << '\n';
127     for (const auto & word : framed_words_left)
128     {
129         cout << word << endl;
130     }
131     cout << string(frame_size, frame_c) << '\n';
132 }
133
134 void FrameWord::display_frame_right() const
135 {
136     short frame_size = max_length_word + 4;
137     cout << string(frame_size, frame_c) << '\n';
138     for (const auto & word : framed_words_right)
139     {
140         cout << word << endl;
141     }
142     cout << string(frame_size, frame_c) << '\n';
143 }
144
145 int main()
146 {
147     cout << "\n\t\tA Box Upon Ye! Program\n\n";
148     bool done = false;
149     vector<string> words;
150     do
151     {
152         char choice;
153         cout << "Are you inputting from keyboard or file?: ";
154         cin >> choice;
155         choice = static_cast<char>(toupper( choice ));
156         cin.ignore(numeric_limits<streamsize>::max(), '\n');
157         if ( choice == 'F' )
158         {
159             ifstream file;
160             string fn;
161             cout << "\nPlease enter the name of your names files: ";
162             getline(cin, fn);
163             file.open(fn);
164             while ( ! file )
165             {
166                 file.close();
167                 file.clear();
168                 cout << "\nIm sorry I could open " << "'" << fn << "'"
169                     << ". Please enter another name: ";
170                 getline(cin, fn);
171                 file.open(fn);
172             }
173             cout << "\nFile " << "'" << fn << "'" << " was opened succesfu
174             string s;
175             while (!file.eof() )
176             {
177                 getline(file, s);
178                 words.push_back(s);

```

```

179         }
180         file.close();
181         file.clear();
182         done = true;
183     }
184     else if ( choice == 'K' )
185     {
186         cout << "\nInsert Sentence: ";
187         string sen;
188         getline(cin, sen);
189         words = word_get(sen);
190         done = true;
191     }
192     else
193     {
194         cout << "You did not input any correct answer."
195             << "Please Try Again\n";
196     }
197
198 } while ( ! done );
199
200 bool done1 = false;
201 bool center = false;
202 bool left = false;
203 bool right = false;
204
205 do
206 {
207     char choice;
208     cout << "\nHow do you want to center the word? (Center, Left, or R:
209     cin >> choice;
210     choice = static_cast<char>(toupper( choice ));
211     cin.ignore(numeric_limits<streamsize>::max(), '\n');
212     if ( choice == 'C' )
213     {
214         center = true;
215         done1 = true;
216     }
217     else if ( choice == 'L' )
218     {
219         left = true;
220         done1 = true;
221     }
222     else if ( choice == 'R' )
223     {
224         right = true;
225         done1 = true;
226     }
227     else
228     {
229         cout << "You did not input Center, Left, or Right."
230             << " Please Try Again";
231     }
232

```

```

233 } while ( ! done1 );
234 char frame_c;
235 bool done2 = false;
236 do
237 {
238     char t;
239     cout << "\nWhat would you like your frame be made of? ";
240     cin >> t;
241     if ( isprint(t))
242     {
243         frame_c = t;
244         done2 = true;
245     }
246     else
247     {
248         cout << "You did not input a printable character."
249             << "Please Try Again";
250     }
251 } while ( ! done2 );
252
253 FrameWord FrameWord(words,frame_c);
254
255 if ( center)
256 {
257     FrameWord.frame_words();
258     cout << "\nFramed Words:\n";
259     FrameWord.display_frame();
260     cout << '\n';
261 }
262 else if (left)
263 {
264     FrameWord.frame_words_left();
265     cout << "\nFramed Words:\n";
266     FrameWord.display_frame_left();
267     cout << '\n';
268 }
269 else if (right)
270 {
271     FrameWord.frame_words_right();
272     cout << "\nFramed Words:\n";
273     FrameWord.display_frame_right();
274     cout << '\n';
275 }
276 return 0;
277 }
278
279 vector<string> word_get(string sen)
280 {
281     vector<string> words;
282     vector<string::size_type> num = find(sen, ' ');
283     string sen1 = sen;
284     for( vector<string::size_type>::size_type p = 0; p != num.size(); ++p)
285     {
286         string::size_type spaces = find_space(sen1, ' ');

```

```

287         if ( spaces != sen1.length())
288         {
289             string word = sen1.substr(0,spaces);
290             words.push_back(word);
291             sen1 = sen1.substr(spaces + 1, sen1.length());
292         }
293     }
294     string word1 = sen.substr(num[num.size() - 1] + 1, sen.length());
295     words.push_back(word1);
296     return words;
297 }

```

bj94684@ares:~\$ CPP strextra box  
box.cpp\*\*\*  
strextra.cpp...  
**box.cpp:** In constructor  
'FrameWord::FrameWord(const  
std::vector<std::\_\_cxx11::basic\_string<char> >&, char)':  
**box.cpp:51:1: warning:**  
'FrameWord::framed\_words' should be initialized  
in the member initialization list [-Weffc++]  
51 | **FrameWord::FrameWord**(const vector<string>  
& inputted\_words, char f\_c)  
| ~~~~~~  
**box.cpp:51:1: warning:**  
'FrameWord::framed\_words\_left'  
should be initialized in the member initialization list  
[-Weffc++]  
**box.cpp:51:1: warning:**  
'FrameWord::framed\_words\_right'  
should be initialized in the member initialization list  
[-Weffc++]

bj94684@ares:~\$ ./box.out

A Box Upon Ye! Program

Are you inputting from keyboard or file?: sa  
You did not input any correct answer.Please Try Again  
Are you inputting from keyboard or file?: key

Insert Sentence: 0 | -+ | /-\

How do you want to center the word? (Center, Left, or Right)? s  
You did not input Center, Left, or Right. Please Try Again  
How do you want to center the word? (Center, Left, or Right)? center

What would you like your frame be made of? X

Framed Words:  
XXXXXXX  
X 0 X  
X | X  
X -+ X

```
X | X
X /-\ X
XXXXXX

bj94684@ares:~$ ./box.out

A Box Upon Ye! Program

Are you inputting from keyboard or file?: keyboard

Insert Sentence: I went to eat food

How do you want to center the word? (Center, Left, or Right)? rightt

What would you like your frame be made of? +

Framed Words:
+++++++
+   I   +
+ went +
+  to  +
+ eat  +
+ food +
+++++++

bj94684@ares:~$ ./box.out

A Box Upon Ye! Program

Are you inputting from keyboard or file?: fi

Please enter the name of your names files: phrase

Im sorry I could open "phrase". Please enter another name: phrase.k

File "phrase.k" was opened succesfully

How do you want to center the word? (Center, Left, or Right)? left

What would you like your frame be made of? ^

Framed Words:
^^^^^^^
^hello ^
^world ^
^      ^
^^^^^^^

bj94684@ares:~$ ./box.out

A Box Upon Ye! Program

Are you inputting from keyboard or file?: fileee
```

```
Please enter the name of your names files: phrasel.k

File "phrasel.k" was opened succesfully

How do you want to center the word? (Center, Left, or Right)? h
You did not input Center, Left, or Right. Please Try Again
How do you want to center the word? (Center, Left, or Right)? right

What would you like your frame be made of? *

Framed Words:
*****
*   the *
*   sky *
*    is *
* falling *
*       *
*****

bj94684@ares:~$ ./box.out

A Box Upon Ye! Program

Are you inputting from keyboard or file?: Key

Insert Sentence: the sky is falling

How do you want to center the word? (Center, Left, or Right)? Center

What would you like your frame be made of? @

Framed Words:
@@@@@@@@@
@ the @
@ sky @
@ is @
@ falling @
@@@@@@@@@

bj94684@ares:~$ exit
exit

Script done on 2023-07-27 12:27:34-05:00 [COMMAND_EXIT_CODE="0"]
```