**ECE 270**



Justin Newman

Quiz #10

Weather Report

October 1, 2014

# Statement of the Problem

The purpose of this program is to utilize the functionality of openFrameworks to query the weather information API (application programming interface) hosted by <http://www.openweathermap.org> for current Temperature, Wind Speed, Cloud Cover Percentage, and Humidity info on 10 cities and print the information to the console window and a text file.

# Description of solution

The data query for each city can be formatted as a URL that the API uses to determine what city you are requesting info for and what format you’d like the information returned to you in as follows:

<http://api.openweathermap.org/data/2.5/weather?q=Detroit&mode=xml>

(City Name) (Format)

The program starts with a series of string arrays which store, respectively, the names of the cities, the URL to be queried for each city, the column headings for the output display, the tags that each piece of information is stored within in the information served by the website, and the attribute corresponding to the desired data in each tab. The program then requests the data for each city in turn, parses the xml served by the website for each piece of info, and displayes it in turn.

# Output and Testing

The testing regimen was very simple for this program, simply run it to see if the information displayed properly and check the information displayed against the <http://www.openweathermap.org> website. The information was correct.

-------------------------------------------------------------------------------

City Temp (k) Clouds Wind (m/s) Humidity

-------------------------------------------------------------------------------

Detroit 276.44 40 2.1 55

New York 274.69 90 3.6 80

Owosso 276.48 90 3.6 84

Chicago 278.48 90 5.1 60

Las Vegas 290.4 1 2.6 27

Madison 276.68 1 3.81 56

Boston 279.75 0 5.7 93

Boulder 290.84 1 1.5 70

Kansas City 285.98 1 4.6 44

New Orleans 286.81 75 4.1 41

-------------------------------------------------------------------------------

# Code

1 #include "ofMain.h"

2 #include "ofApp.h"

3 #include "strings.h"

4

5 **void** getWeatherData(**char** [], **char** []);

6 **void** getAttrString(**char** [], **char** [], **char** [], **char** []);

7 **void** printTabbed(**char** [],FILE \*);

8 **void** printLine(FILE \*);

9

10 **int** main()

11 {

12 FILE \*fp;

13 fp=fopen("output.txt","w");

14 **char** cities[10][12]={"Detroit","New York","Owosso","Chicago",

"Las Vegas","Madison","Boston","Boulder","Kansas City","New Orleans",};

15 **char** urls[10][100]={

"http://api.openweathermap.org/data/2.5/weather?q=Detroit&mode=xml",

"http://api.openweathermap.org/data/2.5/weather?id=5128638&mode=xml",

"http://api.openweathermap.org/data/2.5/weather?q=Owosso&mode=xml",

"http://api.openweathermap.org/data/2.5/weather?q=Chicago&mode=xml",

"http://api.openweathermap.org/data/2.5/weather?id=5506956&mode=xml",

"http://api.openweathermap.org/data/2.5/weather?q=Madison&mode=xml",

"http://api.openweathermap.org/data/2.5/weather?q=Boston&mode=xml",

"http://api.openweathermap.org/data/2.5/weather?q=Boulder&mode=xml",

"http://api.openweathermap.org/data/2.5/weather?id=4273837&mode=xml",

"http://api.openweathermap.org/data/2.5/weather?id=4335045&mode=xml"};

16 **char** labels[5][12]={"City","Temp (k)","Clouds","Wind (m/s)","Humidity"};

17 **char** tags[5][20]={"city","temperature","clouds","wind","humidity"};

18 **char** attribs[5][12]={"name","value","value","speed value","value"};

19 **char** values[6][12];

20 **char** xmlString[1000];

21 **int** i,j;

22

23 printLine(fp);

24

25 **for**(i=0;i<5;i++)

26 {

27 printTabbed(labels[i], fp);

28 }

29

30 printLine(fp);

31

32 **for**(i=0;i<10;i++)

33 {

34 getWeatherData(urls[i],xmlString);

35 **for**(j=0;j<5;j++)

36 {

37 getAttrString(xmlString, tags[j], attribs[j], values[j]);

38 printTabbed(values[j],fp);

39 }

40

41 printf("\n");

42

43

44 }

45

46 printLine(fp);

47

48

49 }

50

51 **void** getWeatherData(**char** urlString[], **char** xmlString[])

52 {

53 **int** n;

54

55 //Retrieving data from internet

56

57 ofHttpResponse resp = ofLoadURL(urlString);

58

59 ofBuffer myBuffer = resp.data;

60

61 **char** \*ptr;

62

63 ptr = myBuffer.getBinaryBuffer();

64

65 n = strlen(ptr);

66

67 **for**(**int** i=0;i<n;i++)

68 {

69 xmlString[i] = \*(ptr + i);

70 }

71

72 xmlString[n] = '\0';

73 }

74

75 **void** getAttrString(**char** xmlString[], **char** tag[], **char** attrName[], **char** valueStr[])

76 {

77 **int** n; //used for string lengths

78

79 **int** done, **count**; //used for while loop

80

81 **char** \*tag\_start, \*tag\_end;

82 **char** \*attrName\_start, \*attrValue\_start;

83

84 //Parsing

85 //Attributes reside inside a certain tag

86 //And attributes come in name-value pairs, as in:

87 //<tag attrName="attrValue"

88 //We need to parse out everything but that attrValue

89

90 //first let's look for that tag

91 tag\_start = strstr(xmlString, tag);

92

93 n = strlen(tag);

94

95 //So the end of the tag name is here--and where the attributes start

96 tag\_end = tag\_start + n;

97

98 //Now, we need to find the attribute

99 attrName\_start = strstr(tag\_end, attrName);

100

101 n = strlen(attrName);

102

103 //now advance past attr\_name

104 attrValue\_start = attrName\_start + n;

105

106 //advance past the = sign and " to the start of attr\_value

107 attrValue\_start = attrValue\_start + 2;

108

109 //now, capture all characters until the closing quote

110

111 **count** = 0;

112 done = 0;

113 **while**(done==0)

114 {

115 **if** (\*(attrValue\_start + **count**)=='"')

116 {

117 done = 1;

118 }

119 **else**

120 {

121 valueStr[**count**] = \*(attrValue\_start + **count**);

122 **count**++;

123 }

124 }

125 valueStr[**count**] = '\0'; //null terminate value\_str

126 }

127

128 //--------------------------------------------------------------

129 /\*void ofApp::setup()

130 {

131 char weatherXML[1000], myValueStr[100];

132

133 //construct the url

134

135 char myUrlString[] = "http://api.openweathermap.org/data/2.5/weather?q=Detroit&mode=xml";

136

137 getWeatherData(myUrlString, weatherXML);

138

139 getAttrString(weatherXML, "temperature", "value", myValueStr);

140

141 printf("\nXML: %s", weatherXML);

142 printf("\nresult: %s", myValueStr);

143 }\*/

144

145 //Prints a string and appends 2 tabs to the end of it, for arranging display

146 **void** printTabbed(**char** stringToPrint[],FILE \*fp)

147 {

148

149 **if**(strlen(stringToPrint)>=8)

150 {

151 printf("%s\t",stringToPrint);

152 fprintf(fp,"%s\t",stringToPrint);

153 }

154 **else**

155 {

156 printf("%s\t\t",stringToPrint);

157 fprintf(fp,"%s\t\t",stringToPrint);

158 }

159 }

160

161 //Prints a line of dashes...

162 **void** printLine(FILE \*fp)

163 {

164 printf("-------------------------------------------------------------------------------\n");

165 fprintf(fp,"-------------------------------------------------------------------------------\n");

166 }

167

168 //--------------------------------------------------------------

169 /\*void ofApp::update()

170 {

171

172 }

173

174 //--------------------------------------------------------------

175 void ofApp::draw()

176 {

177 }

178

179 //--------------------------------------------------------------

180 void ofApp::keyPressed(int key){

181

182 }

183

184 //--------------------------------------------------------------

185 void ofApp::keyReleased(int key){

186

187 }

188

189 //--------------------------------------------------------------

190 void ofApp::mouseMoved(int x, int y){

191

192 }

193

194 //--------------------------------------------------------------

195 void ofApp::mouseDragged(int x, int y, int button){

196

197 }

198

199 //--------------------------------------------------------------

200 void ofApp::mousePressed(int x, int y, int button){

201

202 }

203

204 //--------------------------------------------------------------

205 void ofApp::mouseReleased(int x, int y, int button){

206

207 }

208

209 //--------------------------------------------------------------

210 void ofApp::windowResized(int w, int h){

211

212 }

213

214 //--------------------------------------------------------------

215 void ofApp::gotMessage(ofMessage msg){

216

217 }

218

219 //--------------------------------------------------------------

220 void ofApp::dragEvent(ofDragInfo dragInfo){

221

222 }

223 \*/