C++ NATO ICAO Encoder

Contents

Time required: 30 minutes

- Comment each line of code as shown in the tutorials and other code examples.
- Follow all directions carefully and accurately.
- Think of the directions as minimum requirements.

NATO ICAO Phonetic Alphabet

The North America Treaty Organization (NATO) Phonetic Alphabet is the most widely used spelling alphabet. A spelling alphabet (aka radio alphabet, or telephone alphabet) is a set of words used to stand for the letters of an alphabet in oral communication. Each word in the spelling alphabet typically replaces the name of the letter with which it starts. It is used to spell out words when speaking to someone not able to see the speaker, or when the audio channel is not clear.

The International Civil Aviation Organization (ICAO) Alphabet is a series of words which are used to represent each letter of the alphabet. These are used in critical radio communications between airplanes and ground, and between airplanes in flight to avoid misunderstanding.

To save some time, the map is provided for you to copy and paste into the program.

```
    nato_map['A'] = "Alpha";
    nato_map['B'] = "Bravo";
    nato_map['C'] = "Charlie";
    nato_map['D'] = "Delta";
    nato_map['E'] = "Echo";
    nato_map['F'] = "Foxtrot";
    nato_map['G'] = "Golf";
    nato_map['H'] = "Hotel";
    nato_map['I'] = "India";
    nato_map['J'] = "Juliett";
    nato_map['K'] = "Kilo";
    nato_map['L'] = "Lima";
    nato_map['M'] = "Mike";
    nato_map['N'] = "November";
    nato_map['O'] = "Oscar";
    nato_map['P'] = "Papa";
    nato_map['Q'] = "Quebec";
    nato_map['R'] = "Romeo";
    nato_map['S'] = "Sierra";
    nato_map['T'] = "Tango";
    nato_map['U'] = "Uniform";
    nato_map['V'] = "Victor";
    nato_map['W'] = "Whiskey";
    nato_map['X'] = "X-ray";
    nato_map['Y'] = "Yankee";
    nato_map['Z'] = "Zulu";
```

```cpp
/*
 * Name: nato_encoder.cpp
 * Author:
 * Created:
 * Purpose:
 */

#include <iostream>
#include <unordered_map>

int main()
{
    std::string word;
    // Declare natomap to be of <char, string> type
    // key: char  value: string
    std::unordered_map<char, std::string> nato_map;

    // Use [] method to add keys and values to the unordered_map
    nato_map['A'] = "Alpha";
    nato_map['B'] = "Bravo";
    nato_map['C'] = "Charlie";
    nato_map['D'] = "Delta";
    nato_map['E'] = "Echo";
    nato_map['F'] = "Foxtrot";
    nato_map['G'] = "Golf";
    nato_map['H'] = "Hotel";
    nato_map['I'] = "India";
    nato_map['J'] = "Juliett";
    nato_map['K'] = "Kilo";
    nato_map['L'] = "Lima";
    nato_map['M'] = "Mike";
    nato_map['N'] = "November";
    nato_map['O'] = "Oscar";
    nato_map['P'] = "Papa";
    nato_map['Q'] = "Quebec";
    nato_map['R'] = "Romeo";
    nato_map['S'] = "Sierra";
    nato_map['T'] = "Tango";
    nato_map['U'] = "Uniform";
    nato_map['V'] = "Victor";
    nato_map['W'] = "Whiskey";
    nato_map['X'] = "X-ray";
    nato_map['Y'] = "Yankee";
    nato_map['Z'] = "Zulu";
```

```
46      // Get a single word from user
47      std::cout << "Enter one word only: ";
48      std::cin >> word;
49
50      // Loop through string one character at a time
51      for (int i = 0; i < word.length(); i++)
52      {
53          // Use char key to get string value
54          // Convert each character to upper case for comparison
55          std::cout << nato_map.at(toupper(word[i])) << " " ;
56      }
57      return 0;
58  }
```

Example run:

```
Enter one word only: Bill
Bravo India Lima Lima
```

Assignment Submission

1. Attach the program files.
2. Attach screenshots showing the successful operation of the program.
3. Submit in Blackboard.