

# Hash Tables

May 5, 2020

## 0.1 Dictionary Example in Code

In the cell below, we have created a hash table (`unordered_map`) to store the data from the example above. To create an `unordered_map` in C++, you must include the `<unordered_map>` header, and the syntax for declaring an `unordered_map` is as follows:

```
unordered_map <key_type, value_type> variable_name;
```

In the code below, we check if the key is in the `unordered_map` using the `.find()` method. If the key does not exist in the map, then `.find()` returns an `unordered_map::end()` type. Otherwise, `.find()` returns a [C++ iterator](#), which is a pointer that points to the beginning of the iterable key-value pair.

We haven't covered iterators in this course, and you won't need them for this project, but they are a lot like pointers that can "iterate" forward or backward through a range.

```
In [ ]: #include <iostream>
        #include <vector>
        #include <unordered_map>
        #include <string>
        using std::vector;
        using std::cout;
        using std::unordered_map;
        using std::string;

        int main() {
            // Create strings to use in the hash table.
            string key = "word";
            string def_1 = "a unit of language, consisting of one or more spoken sounds"
                "or their written representation, that functions as a principal carrier"
                "of meaning";
            string def_2 = "speech or talk: to express one's emotion in words";
            string def_3 = "a short talk or conversation: 'Marston, I'd like a word with you.'";
            string def_4 = "an expression or utterance: a word of warning";
            unordered_map <string, vector<string>> my_dictionary;

            // Check if key is in the hash table.
            if (my_dictionary.find(key) == my_dictionary.end()) {
                cout << "The key 'word' is not in the dictionary." << "\n";
```



```

#include<iostream>
#include<vector>
using std::unordered_map;
using std::string;
using std::cout;
using std::vector;

int main(){
    unordered_map<int, string> IDD_codes
    {{972, "Israel"}, {93, "Afghanistan"}, {355, "Albania"}, {213, "Algeria"},
    {376, "Andorra"}, {244, "Angola"}, {54, "Argentina"}, {374, "Armenia"},
    {297, "Aruba"}, {61, "Australia"}, {43, "Austria"}, {994, "Azerbaijan"},
    {973, "Bahrain"}, {880, "Bangladesh"}, {375, "Belarus"}, {32, "Belgium"},
    {501, "Belize"}, {229, "Benin"}, {975, "Bhutan"},
    {387, "Bosnia and Herzegovina"}, {267, "Botswana"}, {55, "Brazil"},
    {246, "British Indian Ocean Territory"}, {359, "Bulgaria"},
    {226, "Burkina Faso"}, {257, "Burundi"}, {855, "Cambodia"},
    {237, "Cameroon"}, {1, "Canada"}, {238, "Cape Verde"},
    {236, "Central African Republic"}, {235, "Chad"}, {56, "Chile"},
    {86, "China"}, {61, "Christmas Island"}, {57, "Colombia"},
    {269, "Comoros"}, {242, "Congo"}, {682, "Cook Islands"},
    {506, "Costa Rica"}, {385, "Croatia"}, {53, "Cuba"}, {537, "Cyprus"},
    {420, "Czech Republic"}, {45, "Denmark"}, {253, "Djibouti"},
    {593, "Ecuador"}, {20, "Egypt"}, {503, "El Salvador"},
    {240, "Equatorial Guinea"}, {291, "Eritrea"}, {372, "Estonia"},
    {251, "Ethiopia"}, {298, "Faroe Islands"}, {679, "Fiji"},
    {358, "Finland"}, {33, "France"}, {594, "French Guiana"},
    {689, "French Polynesia"}, {241, "Gabon"}, {220, "Gambia"},
    {995, "Georgia"}, {49, "Germany"}, {233, "Ghana"}, {350, "Gibraltar"},
    {30, "Greece"}, {299, "Greenland"}, {590, "Guadeloupe"},
    {502, "Guatemala"}, {224, "Guinea"}, {245, "Guinea-Bissau"},
    {595, "Guyana"}, {509, "Haiti"}, {504, "Honduras"}, {36, "Hungary"},
    {354, "Iceland"}, {91, "India"}, {62, "Indonesia"}, {964, "Iraq"},
    {353, "Ireland"}, {972, "Israel"}, {39, "Italy"}, {81, "Japan"},
    {962, "Jordan"}, {254, "Kenya"}, {686, "Kiribati"}, {965, "Kuwait"},
    {996, "Kyrgyzstan"}, {371, "Latvia"}, {961, "Lebanon"}, {266, "Lesotho"},
    {231, "Liberia"}, {423, "Liechtenstein"}, {370, "Lithuania"},
    {352, "Luxembourg"}, {261, "Madagascar"}, {265, "Malawi"},
    {60, "Malaysia"}, {223, "Mali"}, {356, "Malta"},
    {692, "Marshall Islands"}, {596, "Martinique"}, {222, "Mauritania"},
    {230, "Mauritius"}, {262, "Mayotte"}, {52, "Mexico"}, {377, "Monaco"},
    {976, "Mongolia"}, {382, "Montenegro"}, {212, "Morocco"}, {95, "Myanmar"},
    {264, "Namibia"}, {674, "Nauru"}, {977, "Nepal"}, {31, "Netherlands"},
    {599, "Netherlands Antilles"}, {687, "New Caledonia"},
    {64, "New Zealand"}, {505, "Nicaragua"}, {227, "Niger"},
    {234, "Nigeria"}, {683, "Niue"}, {672, "Norfolk Island"},
    {47, "Norway"}, {968, "Oman"}, {92, "Pakistan"},
    {680, "Palau"}, {507, "Panama"}, {675, "Papua New Guinea"},

```

```

{595, "Paraguay"}, {51, "Peru"}, {63, "Philippines"},
{48, "Poland"}, {351, "Portugal"}, {974, "Qatar"},
{40, "Romania"}, {250, "Rwanda"}, {685, "Samoa"},
{378, "San Marino"}, {966, "Saudi Arabia"},
{221, "Senegal"}, {381, "Serbia"}, {248, "Seychelles"},
{232, "Sierra Leone"}, {65, "Singapore"},
{421, "Slovakia"}, {386, "Slovenia"}, {677, "Solomon Islands"},
{27, "South Africa"}, {500, "South Georgia and the South Sandwich Islands"},
{34, "Spain"}, {94, "Sri Lanka"}, {249, "Sudan"},
{597, "Suriname"}, {268, "Swaziland"}, {46, "Sweden"},
{41, "Switzerland"}, {992, "Tajikistan"}, {66, "Thailand"},
{228, "Togo"}, {690, "Tokelau"}, {676, "Tonga"},
{216, "Tunisia"}, {90, "Turkey"}, {993, "Turkmenistan"},
{688, "Tuvalu"}, {256, "Uganda"}, {380, "Ukraine"},
{971, "United Arab Emirates"}, {44, "United Kingdom"}, {1, "United States"},
{598, "Uruguay"}, {998, "Uzbekistan"}, {678, "Vanuatu"},
{681, "Wallis and Futuna"}, {967, "Yemen"}, {260, "Zambia"},
{263, "Zimbabwe"}, {591, "Bolivia, Plurinational State of"},
{673, "Brunei Darussalam"}, {61, "Cocos (Keeling) Islands"},
{243, "Congo, The Democratic Republic of the"}, {225, "Cote d'Ivoire"},
{500, "Falkland Islands (Malvinas)"},
{44, "Guernsey"}, {379, "Holy See (Vatican City State)"},
{852, "Hong Kong"}, {98, "Iran, Islamic Republic of"},
{44, "Isle of Man"}, {44, "Jersey"},
{850, "Korea, Democratic People's Republic of"}, {82, "Korea, Republic of"},
{856, "Lao People's Democratic Republic"},
{218, "Libyan Arab Jamahiriya"}, {853, "Macao"},
{389, "Macedonia, The Former Yugoslav Republic of"},
{691, "Micronesia, Federated States of"},
{373, "Moldova, Republic of"}, {258, "Mozambique"},
{970, "Palestinian Territory, Occupied"}, {872, "Pitcairn"},
{262, "Réunion"}, {7, "Russia"}, {590, "Saint Barthélemy"},
{290, "Saint Helena, Ascension and Tristan Da Cunha"},
{590, "Saint Martin"}, {508, "Saint Pierre and Miquelon"},
{239, "Sao Tome and Principe"}, {252, "Somalia"},
{47, "Svalbard and Jan Mayen"}, {963, "Syrian Arab Republic"},
{886, "Taiwan, Province of China"},
{255, "Tanzania, United Republic of"}, {670, "Timor-Leste"},
{58, "Venezuela, Bolivarian Republic of"}, {84, "Viet Nam"}};
if (IDD_codes.find(960) == IDD_codes.end()) {
    IDD_codes[960] = "Maldives";
}

vector<int> my_codes {1, 55, 86, 960};
for (int code : my_codes) {
    cout << code << ": " << IDD_codes[code] << "\n";
}
}

```

Compile & Execute

See Solution

Loading terminal (id\_8z9gwjg), please wait...