STL Associative Containers: map, unordered map

VGP 131 - Object Oriented Programming in C++ II

Instructor: Ivaldo Tributino

May 23, 2022

ASSIGNMENT INSTRUCTION

- The exam must be submitted by May 29, 2022.
- Each problem presents its own score, the sum of all scores is 100.

Student's Number:
Student's Name:

(10 POINTS) PROBLEM 1

Implement the following operations on maps.

```
//Adds a value with key x and value y to the map.
void add_value(map<int,int> &m,int x,int y)
{
        //Your code here
}

//Returns the value of the key x if present else returns -1

int find_value(map<int,int> &m,int x)
{
    //Your code here
}

// Prints contents of the map ie keys and values
void print_contents(map<int,int> &m)
{
    //Your code here
}
```

```
// Example:
map<int,int> mapFib({{0,0},{1,1},{2,1},{3,2},{4,3},{5,5},{6,8}});

add_value(mapFib, 7, 13 );
cout << find_value(mapFib,7) << endl;
cout << find_value(mapFib,8) << endl;
print_contents(mapFib);

Output:
        13
        -1
        {0 : 0}
        {1 : 1}
        {2 : 1}
        {3 : 2}
        {4 : 3}
        {5 : 5}
        {6 : 8}
        {7 : 13}
```

(10 POINTS) PROBLEM 2

Fill the unordered_map<char, int> counter, with occurrences of each character in the string below (Counts all characters except "␣").

```
string str = "LaSalle␣College␣Vancouver␣has␣several␣scholarships␣for␣
    potential␣students␣to␣apply␣for!␣Check␣out␣if␣you␣qualify."

unordered_map<char, int> counter
```

(10 POINTS) PROBLEM 3

Create two distinct solutions to fix the code below. One modifying only the 1st line and the other modifying only the 2nd line.

```
const map<int, int>& fib = mapFib;
int value = fib[4];
cout << value << endl;

Output:
      3
```

(10 POINTS) PROBLEM 4

The insert_or_assign() function can be considerate a successor of operator[](it improves the operator[])? Justify your answer.

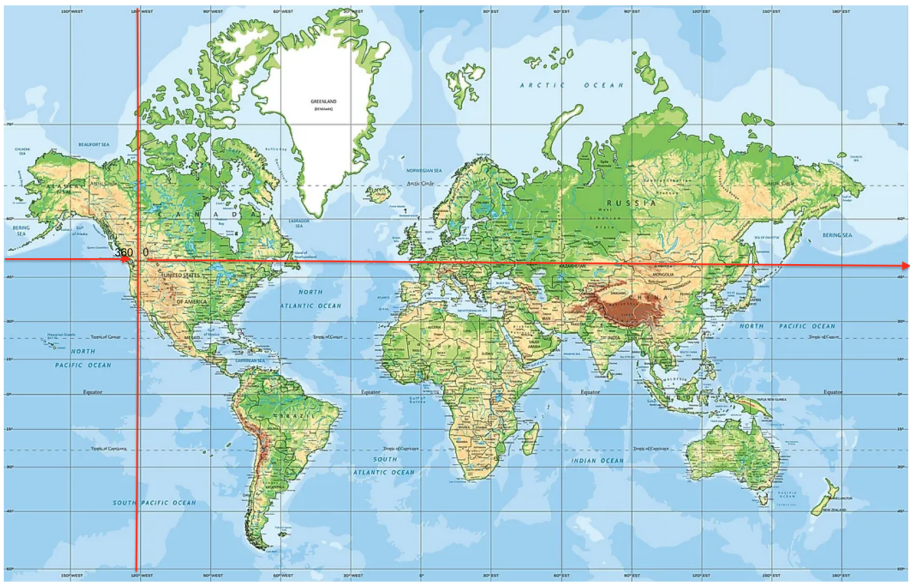
(15 POINTS) PROBLEM 5

Create a container called calculator which is a map that contains pairs of <char, std::function<double(double,double)», where the elements of type char are ('+', '-', '*' and '/') and std :: function <double (double, double) are lambda functions.

(15 POINTS) PROBLEM 6

Implement a data structure that stores information in pairs (key-value), where the index of each value will be stored is calculated using a hash function.

(20 POINTS) PROBLEM 7



Create a class called city where the data members are name, longitude and latitude. Make this class ordered as follows:
Vancouver (49.24966° N, 123.119339° W) will be the smallest of all cities, and the cities will grow as it moves west until surrounds the entire globe (cities on the same meridian will not be considered).

Also create a class called timestamp, where the data member is a unix time and has a member function called unixTimeToDatetime that convert unix time to datetime (yyyy-mm-dd hh:mm:ss).

Finaly, Initialize a map map<city, timestamp> cities with the following data:

```
map<city, timestamp> cities;

cities[city("Toronto", -79.416298, 43.700111)] = timestamp(1489424400);
```

```
cities[city("Vancouver", -123.119339, 49.24966)] = timestamp(1489453200);
cities[city("Dakar", -17.444059, 14.6937)] = timestamp(1489453200);
cities[city("Managua", -86.250397, 12.13282)] = timestamp(1489424400);
cities[city("Nanaimo", -123.936012, 49.16634)] = timestamp(1489435200);
cities[city("Guadalajara", -103.333328, 20.66667)] = timestamp(1489431600);
cities[city("Tokyo", 139.691711, 35.689499)] = timestamp(1489456800);

for(auto &[k,v] : cities){
    cout << k.name << endl;
    cout << v.unixTimeToDatetime() << endl;
}

Expected output:
      Vancouver
      2017-3-14 1:0:0
      Guadalajara
      2017-3-13 19:0:0
      Managua
      2017-3-13 17:0:0
      Toronto
      2017-3-13 17:0:0
      Dakar
      2017-3-14 1:0:0
      Tokyo
      2017-3-14 2:0:0
      Nanaimo
      2017-3-13 20:0:0
```

(10 POINTS) PROBLEM 8

Create a hash function for the city class to enable the following unordered_map.

```
unordered_map<city, timestamp> cities;

cities[city("Toronto", -79.416298, 43.700111)] = timestamp(1489424400);
cities[city("Vancouver", -123.119339, 49.24966)] = timestamp(1489453200);
cities[city("Dakar", -17.444059, 14.6937)] = timestamp(1489453200);
cities[city("Managua", -86.250397, 12.13282)] = timestamp(1489424400);
cities[city("Nanaimo", -123.936012, 49.16634)] = timestamp(1489435200);
cities[city("Guadalajara", -103.333328, 20.66667)] = timestamp(1489431600);
cities[city("Tokyo", 139.691711, 35.689499)] = timestamp(1489456800);
```