**Lab 06**

**Name:** Will Townsend

**Class:** COSC 320 – Section 751

**Date:** 10/13/2020

…No Prelab work…

**Lab Report:**

This lab was straightforward, and I finally understand the iterator objects after working through a few issues I had with which functions return what values. The idea of sets and maps is fairly straightforward and the code in set and map headers was also straightforward. This lab took me approximately 1 hour and 45 minutes (45 minutes total to write all the code for tasks 3 and 4 and an hour to debug). All the code provided is original, 100% by me (beside the d\_state.h functions and overloaded operations) without any help from outside sources.

**Lab:**

**Task 3:**

**d\_state.h:**

#ifndef STATECITY\_CLASS

#define STATECITY\_CLASS

#include <iostream>

#include <string>

using namespace std;

// object stores the state name and city in the state

class stateCity

{

public:

stateCity (const string& name = "", const string& city = ""){

stateName=name;

cityName=city;

}

// output the state and city name in the format

// cityName, stateName

friend ostream& operator<< (ostream& ostr, const stateCity& state){

ostr<<state.cityName<<", "<<state.stateName;

return ostr;

}

// operators < and == must be defined to use with set object.

// operators use only the stateName as the key

friend bool operator< (const stateCity& a, const stateCity& b){

if(a.stateName<b.stateName)

return true;

return false;

}

friend bool operator== (const stateCity& a, const stateCity& b){

if(a.stateName==b.stateName)

return true;

return false;

}

private:

string stateName, cityName;

};

#endif // STATECITY\_CLASS

**lab06\_set.cpp:**

#include<iostream>

#include<stdio.h>

#include<set>

#include<utility>

#include<string>

#include"d\_state.h"

int main(){

stateCity one=stateCity("Maryland","Annapolis");

stateCity two=stateCity("Colorado","Denver");

stateCity three=stateCity("Florida","Miami");

stateCity four=stateCity("Kansas","Kansas City");

stateCity five=stateCity("New Mexico","Roswell");

set<stateCity> s;

s.insert(one);

s.insert(two);

s.insert(three);

s.insert(four);

s.insert(five);

std::string searchState;

puts("Lab 06 (Task 3):");

printf("Enter the state to search for: ");

getline(std::cin,searchState);

std::set<stateCity>::iterator found=s.find(searchState);

if(\*found==searchState)

std::cout<<\*found<<" was found in the set.\n";

else

std::cout<<searchState<<" was not found in the set.\n";

return 0;

}

**Task 4:**

**lab06\_map.cpp:**

#include<iostream>

#include<stdio.h>

#include<map>

#include<utility>

int main(){

std::string state, city;

std::map<std::string,std::string> m;

m.insert(std::pair<std::string,std::string>("Ohio","Columbus"));

m.insert(std::pair<std::string,std::string>("Nevada","Las Vegas"));

m.insert(std::pair<std::string,std::string>("California","Los Angeles"));

m.insert(std::pair<std::string,std::string>("Texas","Dallas"));

m.insert(std::pair<std::string,std::string>("Maryland","Columbia"));

std::string searchState;

puts("Lab 06 (Task 4):");

printf("Enter the state to search for: ");

getline(std::cin,searchState);

std::map<std::string,std::string>::iterator found=m.find(searchState);

if(found->first==searchState)

std::cout<<found->second<<", "<<found->first<<" was found in the map.\n";

else

std::cout<<searchState<<" was not found in the map.\n";

return 0;

}

**Sample Output:**

**Task 3:**

**Output 1:**

Lab 06 (Task 3):

Enter the state to search for: Maryland

Annapolis, Maryland was found in the set.

**Output 2:**

Lab 06 (Task 3):

Enter the state to search for: Colorado

Denver, Colorado was found in the set.

**Output 3:**

Lab 06 (Task 3):

Enter the state to search for: Florida

Miami, Florida was found in the set.

**Output 4:**

Lab 06 (Task 3):

Enter the state to search for: New Mexico

Roswell, New Mexico was found in the set.

**Output 5:**

Lab 06 (Task 3):

Enter the state to search for: Kansas

Kansas City, Kansas was found in the set.

**Output 6:**

Lab 06 (Task 3):

Enter the state to search for: Pennsylvania

Pennsylvania was not found in the set.

**Output 7:**

Lab 06 (Task 3):

Enter the state to search for: Maryladn

Maryladn was not found in the set.

**Task 4:**

**Output 1:**

Lab 06 (Task 4):

Enter the state to search for: California

Los Angeles, California was found in the map.

**Output 2:**

Lab 06 (Task 4):

Enter the state to search for: Texas

Dallas, Texas was found in the map.

**Output 3:**

Lab 06 (Task 4):

Enter the state to search for: Nevada

Las Vegas, Nevada was found in the map.

**Output 4:**

Lab 06 (Task 4):

Enter the state to search for: Ohio

Columbus, Ohio was found in the map.

**Output 5:**

Lab 06 (Task 4):

Enter the state to search for: Maryland

Columbia, Maryland was found in the map.

**Output 6:**

Lab 06 (Task 4):

Enter the state to search for: New Mexico

New Mexico was not found in the map.

**Output 7:**

Lab 06 (Task 4):

Enter the state to search for: Nevaada

Nevaada was not found in the map.