|  |  |  |  |
| --- | --- | --- | --- |
| **Course Name:** | **Data Structures Laboratory (using C/C++)** | **Semester:** | **III** |
| **Date of Performance:** | **10 / 10 / 2023** | **Batch No:** | **A-3** |
| **Faculty Name:** | **Om Goswami** | **Roll No:** | **16014022050** |
| **Faculty Sign & Date:** |  | **Grade/Marks:** | **\_\_\_ / 25** |

**Experiment No: 7**

**Title: Set and Dictionary**

|  |
| --- |
| **Aim and Objective of the Experiment:** |
| Implementation of set and dictionary in C++ using standard template libraries (STL).   1. Implement a set and perform addition/ deletion of elements in it. 2. Implement a dictionary and display its elements with values. |

|  |
| --- |
| **COs to be achieved:** |
| **CO3:** Explain concepts of advanced data structures like set, map dictionary. |

|  |
| --- |
| **Books/Journals/Websites referred:** |
|  |

|  |
| --- |
| **Tools required:** |
| C++ compiler |

|  |
| --- |
| **Theory:** |
| **Sets:** Sets are a type of associative container in which each element has to be unique because the value of the element identifies it. The values are stored in a specific sorted order i.e. either ascending or descending. The std::set class is the part of C++ Standard Template Library (STL) and it is defined inside the <set> header file. Some Basic Functions Associated with Set  * [begin()](https://www.geeksforgeeks.org/setbegin-setend-c-stl/) – Returns an iterator to the first element in the set. * [end()](https://www.geeksforgeeks.org/setbegin-setend-c-stl/) – Returns an iterator to the theoretical element that follows the last element in the set. * [size()](https://www.geeksforgeeks.org/setsize-c-stl/) – Returns the number of elements in the set. * [max\_size()](https://www.geeksforgeeks.org/set-max_size-function-in-c-stl/) – Returns the maximum number of elements that the set can hold. * [empty()](https://www.geeksforgeeks.org/setempty-c-stl/) – Returns whether the set is empty   **Dictionary:** Dictionary is a collection of key, value pairs. They can be changed. In C++, A dictionary can be implemented with the help of Map in standard template libraries (STL). |

|  |
| --- |
| **Implementation details:** |
| 1. **Enlist all the Steps followed and various options explored.** 2. **Explain your program logic and methods used.** 3. **Explain the Importance of the approach followed by you.** |

|  |
| --- |
| **C++ Code implemented:** |
| #include <iostream>  #include <set>  void displaySet(const std::set<int>& mySet) {      std::cout << "Set elements: ";      for (const auto& element : mySet) {          std::cout << element << " ";      }      std::cout << std::endl;  }  int main() {      std::cout << "ketaki mahajan / A-3 / 16014022050";      std::set<int> mySet1;      std::set<int> mySet2;      // Adding elements to the first set      mySet1.insert(5);      mySet1.insert(3);      mySet1.insert(8);      // Adding elements to the second set      mySet2.insert(1);      mySet2.insert(10);      // Displaying elements in the first set      std::cout << "\n\nSet 1 elements: ";      displaySet(mySet1);      // Displaying elements in the second set      std::cout << "Set 2 elements: ";      displaySet(mySet2);      // Swapping the sets      mySet1.swap(mySet2);      // Displaying elements after swapping      std::cout << "Set 1 elements after swapping: ";      displaySet(mySet1);      // Accessing the first element using begin      std::cout << "The first element in Set 1: " << \*mySet1.begin() << std::endl;      return 0;  }  #include <iostream>  #include <map>  int main() {      std::cout << "ketaki mahajan / A-3 / 16014022050";      std::map<std::string, int> myDictionary;      // Adding key-value pairs to the dictionary      myDictionary["apple"] = 5;      myDictionary["banana"] = 3;      myDictionary["cherry"] = 8;      myDictionary["date"] = 1;      // Displaying key-value pairs in the dictionary      std::cout << "\nDictionary elements with values:" << std::endl;      for (const auto& pair : myDictionary) {          std::cout << pair.first << " : " << pair.second << std::endl;      }      // Swap function to swap the values of two keys      std::string key1 = "apple";      std::string key2 = "banana";        if (myDictionary.find(key1) != myDictionary.end() && myDictionary.find(key2) != myDictionary.end()) {          std::swap(myDictionary[key1], myDictionary[key2]);            // Display the dictionary after swapping          std::cout << "Dictionary after swapping values of " << key1 << " and " << key2 << ":" << std::endl;          for (const auto& pair : myDictionary) {              std::cout << pair.first << " : " << pair.second << std::endl;          }      } else {          std::cout << "Keys not found in the dictionary." << std::endl;      }      return 0;  } |

|  |
| --- |
| **Output/ program results after execution:** |
|  |

|  |
| --- |
| **Post Lab Subjective/Objective type Questions:** |
| 1. **Write a C++ program to demonstrate the creation of a descending order set container.**   #include <iostream>  #include <set>  // Custom comparator function for descending order  struct DescendingComparator {      bool operator()(const int& a, const int& b) const {          return a > b;      }  };  int main() {      std::cout << "ketaki mahajan / A-3 / 16014022050";      // Create a set with custom comparator for descending order      std::set<int, DescendingComparator> descendingSet;      // Add elements to the set      descendingSet.insert(90);      descendingSet.insert(52);      descendingSet.insert(19);      descendingSet.insert(88);      descendingSet.insert(30);      descendingSet.insert(27);      descendingSet.insert(18);      descendingSet.insert(44);      // Display elements in descending order      std::cout << "\n\nDescending Order Set Elements: ";      for (const int& element : descendingSet) {          std::cout << element << " ";      }      std::cout << std::endl;      return 0;  } |

|  |
| --- |
| **Conclusion:** |
| We have learned concepts of advanced data structures like set, map dictionary. |

|  |
| --- |
| **Signature of faculty in-charge with Date:** |