**Modern Education Society’s**

**College of Engineering, Pune**

|  |  |
| --- | --- |
| **NAME OF STUDENT:** Prathamesh Kalyan Sable | **CLASS:** SE Comp 1 |
| **SEMESTER/YEAR:** Sem-3 / 2022-23 | **ROLL NO:** 015 |
| **DATE OF PERFORMANCE:**  / /2022 | **DATE OF SUBMISSION:** / /2022 |
| **EXAMINED BY:** Prof. R. H. Shende | **EXPERIMENT NO: C-6** |

###### TITLE: Implemention of SORT AND SEARCH aLGORITHM using stl.

**PROBLEM STATEMENT:**  Write C++ program using STL for sorting and searching user defined records such as personal records (Name, DOB, Telephone number etc) using vector container.

### **OBJECTIVES:**

1. Provide programming insight using OOP constructs.
2. To lay a foundation for STL programming.

### **OUTCOMES:**

1. Develop programming application using object oriented programming language C++.
2. Analyze the strengths of object oriented programming.

**PRE-REQUISITES:**

* 1. Knowledge of template.
  2. Knowledge of Standard Template Library

**APPARATUS:**

Working Computer system with g++ installed

**QUESTIONS:**

1. What is STL and what are the components of it. Explain them.
2. What is Sequence Container and explain the List & Vector in detail with its functions.

**Source Code:**

#include <algorithm>

#include <iomanip>

#include <iostream>

#include <vector>

using namespace std;

struct item {

    int code;

    string name;

    double cost;

    int quantity;

    bool operator==(const item &i2) { return code == i2.code; }

};

// global record list

vector<item> product\_rec;

int sort\_basis;

bool compare(item obj1, item obj2) {

    switch (sort\_basis) {

        case 1:

            return obj1.code < obj2.code;

        case 2:

            return obj1.name < obj2.name;

        case 3:

            return obj1.cost < obj2.cost;

        case 4:

            return obj1.quantity < obj2.quantity;

        default:

            return obj1.code < obj2.code;

    }

}

void insert\_record() {

    struct item i1;

    cout << "Enter Product Code:";

    cin >> i1.code;

    cout << "Enter Product Name:";

    cin.ignore();

    getline(cin, i1.name);

    cout << "Enter Product cost:";

    cin >> i1.cost;

    cout << "Enter Product quantity:";

    cin >> i1.quantity;

    product\_rec.push\_back(i1);

    cout << "Record Inserted sucessfully" << endl;

}

void print\_item(item i) {

    cout << "Product Code is " << i.code << endl;

    cout << "Product Name is " << i.name << endl;

    cout << "Product cost is " << i.cost << endl;

    cout << "Product quantity is " << i.quantity << endl;

}

void display() {

cout << setw(5) << "Code " << " " << setw(15) << "Name " << " " << setw(7) << "Cost " << " " << setw(5) << "Quantity " << endl;

    for (int i = 0; i < product\_rec.size(); i++) {

        cout << setw(5) << left << product\_rec[i].code << " ";

        cout << setw(15) << product\_rec[i].name.substr(0, 15) << " ";

        cout << setw(7) << product\_rec[i].cost << " ";

        cout << setw(5) << product\_rec[i].quantity << endl;

    }

}

void sort\_product() {

    cout << "Sort by 1. Code 2. Name 3.Cost 4.Quantity" << endl;

    cout << "Enter your choice:";

    cin >> sort\_basis;

    sort(product\_rec.begin(), product\_rec.end(), compare);

    cout << "\nSorted Products are:" << endl;

    display();

    cout << endl;

}

void search\_product() {

    item i1;

    cout << "Enter Product Code to search:";

    cin >> i1.code;

    vector<item>::iterator ans = find(product\_rec.begin(), product\_rec.end(), i1);

    if (ans == product\_rec.end() && !(i1 == \*ans)) {

        cout << "Product Not Found" << endl;

    } else {

        cout << "Product details are" << endl;

        print\_item(\*ans);

    }

}

int main() {

    int ch;

    do {

        cout << "--: MENU :--" << endl;

        cout << "1. Insert Record" << endl;

        cout << "2. Sort Records" << endl;

        cout << "3. Search Record" << endl;

        cout << "4. Display All Records" << endl;

        cout << "5. Exit" << endl;

        cout << "Enter Your choice:";

        cin >> ch;

        switch (ch) {

            case 1:

                insert\_record();

                break;

            case 2:

                sort\_product();

                break;

            case 3:

                search\_product();

                break;

            case 4:

                display();

                break;

            case 5:

                cout << "Thank you for using App" << endl;

                break;

            default:

                cout << "Enter a valid choice" << endl;

                break;

        }

    } while (ch != 5);

    return 0;

}

**Output:**





