#include <iostream>

#include <vector>

#include <string>

Class Contact {

Private:

Std::string name;

Std::string phoneNumber;

Public:

// Constructor to initialize contact

Contact(std::string name, std::string phoneNumber) {

This->name = name;

This->phoneNumber = phoneNumber;

}

// Function to get the name of the contact

Std::string getName() const {

Return name;

}

// Function to get the phone number of the contact

Std::string getPhoneNumber() const {

Return phoneNumber;

}

// Function to display contact details

Void displayContact() const {

Std::cout << “Name: “ << name << “, Phone Number: “ << phoneNumber << std::endl;

}

};

Class ContactManager {

Private:

Std::vector<Contact> contacts;

Public:

// Function to add a new contact

Void addContact(const std::string& name, const std::string& phoneNumber) {

Contacts.emplace\_back(name, phoneNumber);

Std::cout << “Contact added successfully!” << std::endl;

}

// Function to view all contacts

Void viewContacts() const {

If (contacts.empty()) {

Std::cout << “No contacts available.” << std::endl;

Return;

}

Std::cout << “List of Contacts:” << std::endl;

For (const auto& contact : contacts) {

Contact.displayContact();

}

}

// Function to delete a contact by name

Void deleteContact(const std::string& name) {

Auto it = std::remove\_if(contacts.begin(), contacts.end(),

[&name](const Contact& contact) {

Return contact.getName() == name;

});

If (it != contacts.end()) {

Contacts.erase(it, contacts.end());

Std::cout << “Contact deleted successfully!” << std::endl;

} else {

Std::cout << “Contact not found.” << std::endl;

}

}

};

Void displayMenu() {

Std::cout << “Contact Management System” << std::endl;

Std::cout << “1. Add Contact” << std::endl;

Std::cout << “2. View Contacts” << std::endl;

Std::cout << “3. Delete Contact” << std::endl;

Std::cout << “4. Exit” << std::endl;

}

Int main() {

ContactManager manager;

Int choice;

Std::string name, phoneNumber;

Do {

displayMenu();

std::cout << “Enter your choice: “;

std::cin >> choice;

switch (choice) {

case 1:

std::cout << “Enter Name: “;

std::cin.ignore();

std::getline(std::cin, name);

std::cout << “Enter Phone Number: “;

std::getline(std::cin, phoneNumber);

manager.addContact(name, phoneNumber);

break;

case 2:

manager.viewContacts();

break;

case 3:

std::cout << “Enter Name of Contact to Delete: “;

std::cin.ignore();

std::getline(std::cin, name);

manager.deleteContact(name);

break;

case 4:

std::cout << “Exiting the program.” << std::endl;

break;

default:

std::cout << “Invalid choice. Please try again.” << std::endl;

}

Std::cout << std::endl;

} while (choice != 4);

Return 0;

}