#include <iostream>

#include <vector>

#include <string>

#include <limits> // Required for numeric\_limits

using namespace std;

// Structure to represent a task

struct Task {

string description;

bool completed;

Task(const string& desc) : description(desc), completed(false) {}

};

// Function to get string input from the user

string getStringInput(const string& prompt) {

string input;

cout << prompt;

getline(cin >> ws, input);

return input;

}

// Function to get integer input from the user

int getIntegerInput(const string& prompt) {

int input;

while (true) {

cout << prompt;

cin >> input;

if (cin.fail()) {

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << "Invalid input. Please enter a number.\n";

} else {

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

break;

}

}

return input;

}

// Function to add a new task

void addTask(vector<Task>& tasks) {

string description = getStringInput("Enter task description: ");

tasks.emplace\_back(description);

cout << "Task added successfully.\n";

}

// Function to view all tasks

void viewTasks(const vector<Task>& tasks) {

if (tasks.empty()) {

cout << "No tasks to display.\n";

return;

}

cout << "\n--- To-Do List ---\n";

for (size\_t i = 0; i < tasks.size(); ++i) {

cout << i + 1 << ". [" << (tasks[i].completed ? "X" : " ") << "] " << tasks[i].description << "\n";

}

cout << "--------------------\n";

}

// Function to mark a task as completed

void markCompleted(vector<Task>& tasks) {

if (tasks.empty()) {

cout << "No tasks to mark as completed.\n";

return;

}

viewTasks(tasks);

int taskNumber = getIntegerInput("Enter the number of the task to mark as completed: ");

if (taskNumber >= 1 && taskNumber <= tasks.size()) {

tasks[taskNumber - 1].completed = true;

cout << "Task marked as completed.\n";

} else {

cout << "Invalid task number.\n";

}

}

// Function to delete a task

void deleteTask(vector<Task>& tasks) {

if (tasks.empty()) {

cout << "No tasks to delete.\n";

return;

}

viewTasks(tasks);

int taskNumber = getIntegerInput("Enter the number of the task to delete: ");

if (taskNumber >= 1 && taskNumber <= tasks.size()) {

tasks.erase(tasks.begin() + taskNumber - 1);

cout << "Task deleted successfully.\n";

} else {

cout << "Invalid task number.\n";

}

}

int main() {

vector<Task> tasks;

int choice;

do {

cout << "\nTo-Do List Menu:\n";

cout << "1. Add Task\n";

cout << "2. View Tasks\n";

cout << "3. Mark Task as Completed\n";

cout << "4. Delete Task\n";

cout << "5. Exit\n";

cout << "Enter your choice: ";

choice = getIntegerInput("Enter your choice: ");

switch (choice) {

case 1:

addTask(tasks);

break;

case 2:

viewTasks(tasks);

break;

case 3:

markCompleted(tasks);

break;

case 4:

deleteTask(tasks);

break;

case 5:

cout << "Exiting program.\n";

break;

default:

cout << "Invalid choice. Please try again.\n";

}

} while (choice != 5);

return 0;

}