**LAB TASK NO 3**

**QUESTION NO 01:**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**class Wallet {**

**string owner;**

**double balance;**

**string history[100];**

**int index = 0;**

**public:**

**Wallet(string name, double initialBalance) {**

**owner = name;**

**balance = initialBalance;**

**}**

**void deposit(double amount) {**

**if (amount <= 0) {**

**cout << "Amount must be positive." << endl;**

**return;**

**}**

**balance += amount;**

**cout << owner << " deposited " << amount << "." << endl;**

**history[index++] = "Deposited: " + to\_string(amount);**

**}**

**void withdraw(double amount) {**

**if (amount <= 0) {**

**cout << "Amount must be positive." << endl;**

**return;**

**}**

**if (amount > balance) {**

**cout << "Not enough balance." << endl;**

**return;**

**}**

**balance -= amount;**

**cout << owner << " spent " << amount << "." << endl;**

**history[index++] = "Spent: " + to\_string(amount);**

**}**

**void showHistory() {**

**cout << owner << "'s Transaction History:" << endl;**

**if (index == 0) {**

**cout << "No transactions yet." << endl;**

**} else {**

**for (int i = 0; i < index; i++) {**

**cout << history[i] << endl;**

**}**

**}**

**}**

**void checkBalance() {**

**cout << "Balance for " << owner << ": " << balance << endl;**

**}**

**void alertLowBalance(double limit) {**

**if (balance < limit) {**

**cout << "Warning: Balance is below " << limit << endl;**

**} else {**

**cout << "Balance is above " << limit << endl;**

**}**

**}**

**};**

**int main() {**

**Wallet userWallet("Saad", 7000);**

**userWallet.deposit(500);**

**userWallet.withdraw(5000);**

**userWallet.showHistory();**

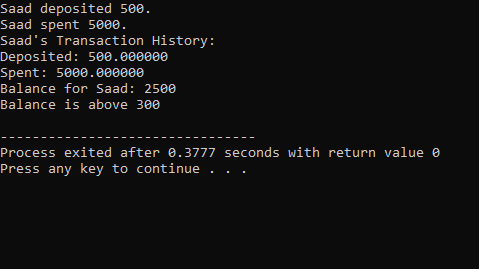
**userWallet.checkBalance();**

**userWallet.alertLowBalance(300);**

**return 0;**

**}**

**Output:**

****

**QUESTION NO 2:**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**class ActivityTracker {**

**string userName;**

**int totalSteps;**

**int stepGoal;**

**float calories;**

**public:**

**ActivityTracker(string uname, int dailyGoal) {**

**userName = uname;**

**stepGoal = dailyGoal;**

**totalSteps = 0;**

**calories = 0;**

**}**

**void addSteps(int newSteps) {**

**if (newSteps <= 0) {**

**cout << "Invalid step count." << endl;**

**return;**

**}**

**totalSteps += newSteps;**

**calories += newSteps \* 0.04;**

**cout << "Steps added: " << newSteps << ", Total steps: " << totalSteps**

**<< ", Calories burned: " << calories << endl;**

**}**

**void evaluateGoal() {**

**if (totalSteps >= stepGoal) {**

**cout << "Great job! You achieved your daily goal!" << endl;**

**} else {**

**cout << "You need " << (stepGoal - totalSteps) << " more steps to reach your goal." << endl;**

**}**

**}**

**void showSummary() {**

**cout << "--- Fitness Summary ---" << endl;**

**cout << "User: " << userName << endl;**

**cout << "Steps Taken: " << totalSteps << endl;**

**cout << "Calories Burned: " << calories << endl;**

**evaluateGoal();**

**cout << "-----" << endl;**

**}**

**};**

**int main() {**

**ActivityTracker user1("Alishba", 10000);**

**user1.addSteps(100);**

**user1.showSummary();**

**user1.addSteps(4000);**

**user1.showSummary();**

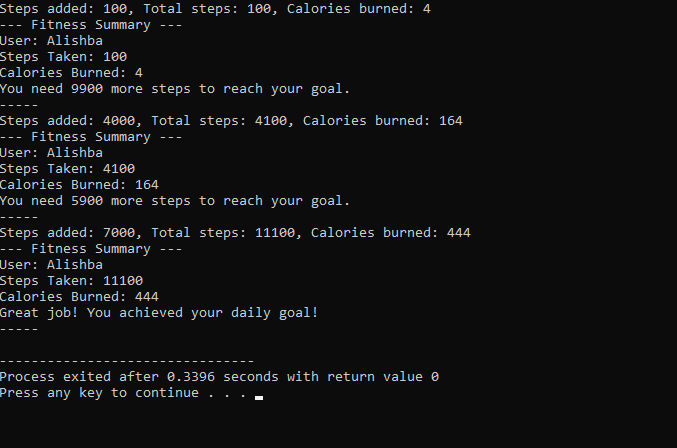
**user1.addSteps(7000);**

**user1.showSummary();**

**return 0;**

**}**

**Output:**

****

**Question no 3:**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**class BookStore {**

**string bookList[100];**

**int stock[100];**

**string borrower[100];**

**int totalBooks;**

**public:**

**BookStore() {**

**totalBooks = 0;**

**}**

**void addBook(string title) {**

**if (totalBooks < 100) {**

**bookList[totalBooks] = title;**

**stock[totalBooks] = 1;**

**borrower[totalBooks] = "";**

**cout << "Book '" << title << "' added to the store.\n";**

**totalBooks++;**

**} else {**

**cout << "No more space in the store.\n";**

**}**

**}**

**void lendBook(string title, string person) {**

**for (int i = 0; i < totalBooks; i++) {**

**if (bookList[i] == title && stock[i] == 1) {**

**stock[i] = 0;**

**borrower[i] = person;**

**cout << "'" << title << "' borrowed by " << person << ".\n";**

**return;**

**}**

**if (bookList[i] == title && stock[i] == 0) {**

**cout << "'" << title << "' is already borrowed by " << borrower[i] << ".\n";**

**return;**

**}**

**}**

**cout << "Book not found in the store.\n";**

**}**

**void showBooks() {**

**cout << "\n----- BOOKSTORE INVENTORY -----\n";**

**for (int i = 0; i < totalBooks; i++) {**

**cout << "Title: " << bookList[i] << endl;**

**if (stock[i] == 0) {**

**cout << "Status: Lent Out\n";**

**cout << "Borrower: " << borrower[i] << endl;**

**} else {**

**cout << "Status: Available\n";**

**}**

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

**}**

**}**

**};**

**int main() {**

**BookStore store;**

**store.addBook("Inferno");**

**store.addBook("Da Vinci Code");**

**store.addBook("Angels and Demons");**

**store.showBooks();**

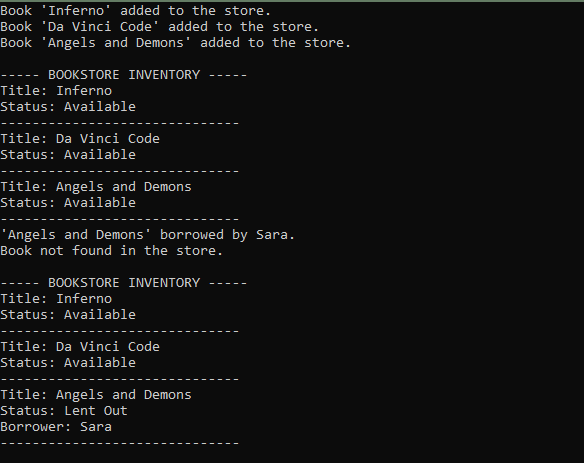
**store.lendBook("Angels and Demons", "Sara");**

**store.lendBook("The Lost Symbol", "Ali");**

**store.showBooks();**

**return 0;**

**}**

**Output:  
**

**Question 4:**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**class FuelManager {**

**string make;**

**string variant;**

**float tankCapacity;**

**float currentFuel;**

**float mileage;**

**public:**

**FuelManager(string cMake, string cVariant, float capacity, float avg) {**

**make = cMake;**

**variant = cVariant;**

**tankCapacity = capacity;**

**mileage = avg;**

**currentFuel = capacity;**

**}**

**void checkFuel() {**

**if (currentFuel <= 0.1 \* tankCapacity) {**

**cout << "Low fuel! Refuel soon.\n";**

**}**

**}**

**void travel(float km) {**

**float fuelNeeded = km \* mileage;**

**if (fuelNeeded > currentFuel) {**

**cout << "Insufficient fuel for the trip.\n";**

**} else {**

**currentFuel -= fuelNeeded;**

**cout << "Distance covered: " << km << " km. Fuel left: " << currentFuel << " liters.\n";**

**checkFuel();**

**}**

**}**

**void addFuel(float liters) {**

**if (liters <= 0) {**

**cout << "Invalid fuel amount.\n";**

**} else if (currentFuel + liters > tankCapacity) {**

**currentFuel = tankCapacity;**

**cout << "Tank is now full.\n";**

**} else {**

**currentFuel += liters;**

**cout << "Refueled. Current fuel: " << currentFuel << " liters.\n";**

**}**

**}**

**void showDetails() {**

**cout << "\n--- VEHICLE INFO ---\n";**

**cout << "Make: " << make << endl;**

**cout << "Model: " << variant << endl;**

**cout << "Fuel Capacity: " << tankCapacity << " liters\n";**

**cout << "Current Fuel: " << currentFuel << " liters\n";**

**cout << "Mileage: " << mileage << " liters/km\n";**

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

**}**

**};**

**int main() {**

**FuelManager vehicle("Toyota", "Corolla", 50, 0.06);**

**vehicle.showDetails();**

**vehicle.travel(400);**

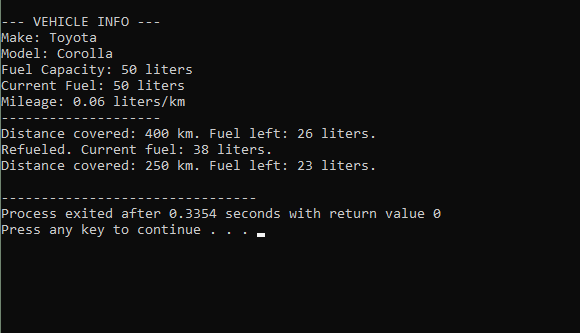
**vehicle.addFuel(12);**

**vehicle.travel(250);**

**return 0;**

**}**

**Output:**

****

**Question 5:**

**#include <iostream>**

**#include <string>**

**using namespace std;**

**class MusicManager {**

**string tracks[100];**

**int totalTracks;**

**string nowPlaying;**

**public:**

**MusicManager() {**

**totalTracks = 0;**

**}**

**void addTrack(string track) {**

**if (totalTracks < 100) {**

**tracks[totalTracks] = track;**

**totalTracks++;**

**cout << "Track added: " << track << endl;**

**} else {**

**cout << "Playlist is full." << endl;**

**}**

**}**

**void removeTrack(string track) {**

**int pos = -1;**

**for (int i = 0; i < totalTracks; i++) {**

**if (tracks[i] == track) {**

**pos = i;**

**break;**

**}**

**}**

**if (pos != -1) {**

**for (int i = pos; i < totalTracks - 1; i++) {**

**tracks[i] = tracks[i + 1];**

**}**

**totalTracks--;**

**cout << "Removed: " << track << endl;**

**} else {**

**cout << "Track not found." << endl;**

**}**

**}**

**void playTrack(string track) {**

**bool found = false;**

**for (int i = 0; i < totalTracks; i++) {**

**if (tracks[i] == track) {**

**nowPlaying = track;**

**found = true;**

**break;**

**}**

**}**

**if (found) {**

**cout << "Playing: " << nowPlaying << endl;**

**} else {**

**cout << "Track not found." << endl;**

**}**

**}**

**void showPlaylist() {**

**if (totalTracks == 0) {**

**cout << "Playlist is empty." << endl;**

**} else {**

**cout << "--- Playlist ---" << endl;**

**for (int i = 0; i < totalTracks; i++) {**

**cout << (i + 1) << ". " << tracks[i] << endl;**

**}**

**}**

**}**

**};**

**int main() {**

**MusicManager m;**

**m.addTrack("Track 1");**

**m.addTrack("Track 2");**

**m.addTrack("Track 3");**

**m.showPlaylist();**

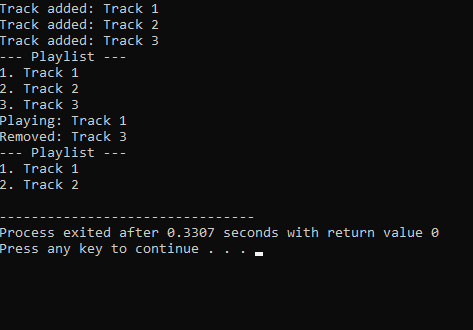
**m.playTrack("Track 1");**

**m.removeTrack("Track 3");**

**m.showPlaylist();**

**return 0;**

**}**

**Output:  
**