

TEB1013 - STRUCTURED PROGRAMMING

```

#include <iostream> // Include the input/output library
#include <fstream> // Include the file handling library
#include <string.h> // Include the string manipulation library

using namespace std; // Use the standard namespace

//Declare functions
void resetCurrentRecord();
void readCurrentRecord();
void login();
void registration();
void forgot();
void userMenu();
void AdminMenu();
void roomBooking();
void amenitiesRequest();
void checkBill();
void viewRooms();
void viewAmenities();
void viewPrices();
void salesReport();

// Declare global variables
int single = 65, twin = 100;
int Qsingle = 20, Qtwin = 40, Qtowel = 75, Qwater = 100, Qbiscuit = 100, Qpillow = 75;
int quant, choice, total_room = 0;

// Declare variables to track the total price for each item
int Total_rooms = 0, Total_single = 0, Total_twin = 0, Total_towel = 0, Total_water = 0, Total_biscuit = 0, Total_pillow = 0;

// Declare variable to track sold items
int Ssingle, Stwin, Stowel, Swater, Sbiscuit, Spillow;

int count = 0; // A variable to track login attempts
string userID, password, id, pass; // Strings to store user credentials

int main()
{
    readCurrentRecord(); // Read the current record from the file
    int mainmenuchoice;
    cout << "\t\t _____ One Night Hotel" _____ \n\n";
    cout << "\t| Press 1 to REGISTER " << endl;
    cout << "\t| Press 2 to LOGIN" << endl;
    cout << "\t| Press 3 if you forgot your password" << endl;
    cout << "\t| Press 4 to EXIT" << endl;
    cout << "\n\t\t Please enter your choice: ";
    cin >> mainmenuchoice; // Read the user's choice from the input
    cout << endl;

    switch (mainmenuchoice)
    {
        case 1:
            registration(); // Call the registration function
            break;
        case 2:
            login(); // Call the login function
            break;
        case 3:
            forgot(); // Call the password recovery function
            break;
        case 4:
            system("cls");
            cout << "\t\t\t Thank you! \n\n"; // Display a farewell message
            exit(0); // Terminate the programme
        default:
            cout << "\t\t\t Invalid choice. Please select from the options given above \n" << endl;
            main(); // Call the main function recursively
    }
    return 0;
}

void login()
{
    system("cls");
    // Prompt the user to enter their credentials
    cout << "\t Enter your User ID: ";
    cin >> userID;
    cout << "\t Enter your Password: ";
    cin >> password;

    // Check if the user ID and password match the stored credentials
    if(userID == "admin69" && password == "beverlyHills178"){

```

```

cout << "\n\t\t\t Login admin successful! \n\n";
AdminMenu(); // Call the admin menu function
}
else {
    ifstream input("credentials.txt"); // Open the file for reading user records
    string line;
    bool found = false;
    while (getline(input, line)) {
        string credentials[2];
        int i = 0;
        size_t pos = 0;
        string token;
        while ((pos = line.find(",")) != string::npos) {
            token = line.substr(0, pos);
            credentials[i] = token;
            line.erase(0, pos + 1);
            i++;
        }
        credentials[i] = line;
        if (credentials[0] == userID && credentials[1] == password) {
            found = true;
            break;
        }
    }
    input.close(); // Close the file

    if (found) {
        count = 1; // Set the login attempt count to 1
        system("cls"); // Clear the screen
        cout << userID << "\n Your LOGIN is successful \n Thanks for logging in! \n\n";
        userMenu(); // Call the main menu function for regular users
    }
    else {
        system("cls");
        cout << "\n LOGIN ERROR \n Please check your username and password \n\n";
        main(); // Return to the main menu if login fails
    }
}

void registration()
{
    system("cls");
    // Prompt the user to enter their desired credentials
    cout << "\t\t\t Enter your desired User ID: ";
    cin >> id;
    cout << "\t\t\t Enter your desired Password: ";
    cin >> pass;

    // Store the user's credentials in a file
    ofstream file;
    file.open("credentials.txt", ios::app);
    file << id << "," << pass << endl;
    file.close();

    cout << "\n\t\t\t Registration successful! \n\n";
    login(); // Call the login function to log the user in after registration
}

void forgot()
{
    system("cls");
    // Prompt the user to enter their user ID
    cout << "\t\t\t Enter your User ID: ";
    cin >> userID;

    // Search for the user's credentials in the file
    ifstream file;
    file.open("credentials.txt");
    bool found = false;
    string line;
    while (getline(file, line)) {
        if (line.find(userID) != string::npos) {
            found = true;
            break;
        }
    }
    file.close();

    // If the user's credentials are found, prompt them to enter a new password
    if (found) {
        cout << "\t\t\t Enter your new Password: ";
    }
}

```

```

    cin >> password;

    // Update the user's password in the file
    file.open("credentials.txt");
    ofstream temp;
    temp.open("temp.txt");
    while(getline(file, line)){
        if(line.find(userID) != string::npos){
            temp << userID << "," << password << endl;
        }
        else{
            temp << line << endl;
        }
    }
    file.close();
    temp.close();
    remove("credentials.txt");
    rename("temp.txt", "credentials.txt");

    cout << "\n\t\t\t Password updated successfully! \n\n";
    login(); // Call the login function to log the user in with the new password
}
else{
    cout << "\n\t\t\t User ID not found. Please try again. \n\n";
    forgot(); // Call the forgot function recursively
}
}

void userMenu()
{
    system("cls");

    int choice;
    cout << "\n\n\t\t\t User Menu \n\n";
    cout << "1. Room Booking\n2. Amenities Request\n3. Check Bill\n4. Logout" << endl;
    cout << "Action: ";
    cin >> choice;
    switch (choice)
    {
        case 1:
            roomBooking();
            break;
        case 2:
            amenitiesRequest();
            break;
        case 3:
            checkBill();
            break;
        case 4:
            cout << "\t\t\t Logging out... \n\n"; // Display a logout message
            system("cls");
            main();
            break;
        default:
            cout << "Invalid choice. Please try again." << endl;
            userMenu();
    }
}

void roomBooking()
{
    system("cls");
    cout << "\t\t\t _____ Room Booking _____ \n\n";

    // Read the current_record.txt file to check for room availability
    ifstream current_record("current_record.txt");
    bool single_available = false;
    bool twin_available = false;
    string firstline;
    string secondline;

    // Read the first line (Single Rooms)
    if (getline(current_record, firstline)){
        int currentSingle = stoi(firstline); // Extract the current stock count for Single Rooms
        if(currentSingle > 0){
            single_available = true;
        }
    }
    // Read the second line (Twin Rooms)
    if (getline(current_record, secondline)){
        int currentTwin = stoi(secondline); // Extract the current stock count for Twin Rooms
    }
}

```



```

// Check if the user has booked a room by looking for their bill report file
string bill_file_name = userID + "_bill_report.txt";
ifstream bill_file(bill_file_name);
if (!bill_file)
{
    cout << "\n\n\t\t\t\t\t\t\t You must book a room before making an amenities request. \n\n";
    userMenu();
    return;
}
bill_file.close();

// Prompt the user to select an amenity
int amenity_choice;
cout << "Select an amenity: \n";
cout << "1. Towel (RM0)\n";
cout << "2. Water (RM2)\n";
cout << "3. Biscuit (RM6)\n";
cout << "4. Pillow (RM0)\n";
cout << "Action: ";
cin >> amenity_choice;

// Add the amenity to the bill report file
ofstream bill_file_out(bill_file_name, ios::app);
if (amenity_choice == 1)
{
    bill_file_out << "Towel: RM0\n";
    Total_towel += 0;
    Stowel++;
    Qtowel--;
}
else if (amenity_choice == 2)
{
    bill_file_out << "Water: RM2\n";
    Total_water += 2;
    Swater++;
    Qwater--;
}
else if (amenity_choice == 3)
{
    bill_file_out << "Biscuit: RM6\n";
    Total_biscuit += 6;
    Sbiscuit++;
    Qbiscuit--;
}
else if (amenity_choice == 4)
{
    bill_file_out << "Pillow: RM0\n";
    Total_pillow += 0;
    Spillow++;
    Qpillow--;
}
else
{
    cout << "\n\n\t\t\t\t\t\t\t Invalid choice. \n\n";
    return;
}
bill_file_out << "\n";
bill_file_out.close();

// Update stocks in current_record.txt
ofstream current_record_file("current_record.txt", ios::trunc);
current_record_file << Qsingle << endl;
current_record_file << Qtwin << endl;
current_record_file << Qtowel << endl;
current_record_file << Qwater << endl;
current_record_file << Qbiscuit << endl;
current_record_file << Qpillow << endl;
current_record_file.close();

// Display a success message and return to the user menu
cout << "\n\n\t\t\t\t\t\t\t Amenities request made successfully. \n\n";

userMenu();
}

void checkBill()
{
    system("cls");
    cout << "\t\t\t _____" << userID << "'s Bill _____ \n\n";
}

// Check if the user has a bill report file

```

```

string bill_file_name = userID + "_bill_report.txt";
ifstream bill_file(bill_file_name);
if (!bill_file)
{
    cout << "\n\n\t\t\t\t\t You have no bill to check. \n\n";
    // Prompt user whether they want go to user menu or exit programme
    int choice;
    cout << "Do you want to exit? \n1. Yes\n2. No" << endl << "Action: ";
    cin >> choice;
    if (choice == 1)
    {
        cout << "\t\t\t Logging out... \n\n"; // Display a logout message
        main();
    }
    else
    {
        system("cls");
        userMenu();
    }
}

// Read the contents of the file and extract the total amount spent
string line;
int total_spending = 0;
while (getline(bill_file, line))
{
    if (line.find("RM") != string::npos)
    {
        // Extract the total amount spent from the line
        int pos = line.find("RM");
        string amount_str = line.substr(pos + 2);
        int amount = stoi(amount_str);
        total_spending += amount;
    }
    cout << line << endl;
}
bill_file.close();

// Display the total amount spent to the console
cout << "\n\t\t\t\t\t Total Spending: RM" << total_spending << endl;

// Prompt user whether they want go to user menu or exit programme
int choice;
cout << "\nDo you want to exit? \n1. Yes\n2. No" << endl << "Action: ";
cin >> choice;
if (choice == 1)
{
    cout << "\t\t\t Logging out... \n\n"; // Display a logout message
    main();
}
else
{
    system("cls");
    userMenu();
}

void readCurrentRecord()
{
    ifstream file("current_record.txt"); // Open the file for reading

    if(!file.is_open()){
        // If the file does not exist, create it and initialize the quantities
        ofstream newFile("current_record.txt");
        int quantities[] = {20, 40, 75, 100, 100, 75};
        string items[] = {"Single Rooms", "Twin Rooms", "Towels", "Bottles of Water", "Biscuits", "Pillows"};
        for(int i = 0; i < 6; i++){
            newFile << quantities[i] << endl;
        }
        newFile.close();
        // Copy the quantities to the global variables
        for(int i = 0; i < 6; i++){
            switch(i){
                case 0:
                    Qsingle = quantities[i];
                    Ssingle = 0;
                    break;
                case 1:
                    Qtwin = quantities[i];
                    Stwin = 0;
                    break;
                case 2:

```

```

        Qtowel = quantities[i];
        Stowel = 0;
        break;
    case 3:
        Qwater = quantities[i];
        Swater = 0;
        break;
    case 4:
        Qbiscuit = quantities[i];
        Sbiscuit = 0;
        break;
    case 5:
        Qpillow = quantities[i];
        Spillow = 0;
        break;
    }
}
}

else{
    // If the file exists, read the quantities from the file
    int quantities[6];
    string items[6] = {"Single Rooms", "Twin Rooms", "Towels", "Bottles of Water", "Biscuits", "Pillows"};
    for(int i = 0; i < 6; i++){
        file >> quantities[i];
    }
    file.close();
    // Copy the quantities to the global variables
    for(int i = 0; i < 6; i++){
        switch(i){
            case 0:
                Qsingle = quantities[i];
                Ssingle = 20 - Qsingle;
                break;
            case 1:
                Qtwin = quantities[i];
                Stwin = 40 - Qtwin;
                break;
            case 2:
                Qtowel = quantities[i];
                Stowel = 75 - Qtowel;
                break;
            case 3:
                Qwater = quantities[i];
                Swater = 100 - Qwater;
                break;
            case 4:
                Qbiscuit = quantities[i];
                Sbiscuit = 100 - Qbiscuit;
                break;
            case 5:
                Qpillow = quantities[i];
                Spillow = 75 - Qpillow;
                break;
        }
    }
}
}

void resetCurrentRecord()
{
    system("cls");
    cout << "\t\t\t _____ Reset Current Record _____ \n\n\n";
    cout << "Enter the date of reset (DD-MM-YYYY) separate with hyphen (-): ";
    string date;
    cin >> date;

    // Rename the current_record.txt file to [Date]_record.txt
    string old_file_name = "current_record.txt";
    string new_file_name = date + "_record.txt";
    if (rename(old_file_name.c_str(), new_file_name.c_str()) != 0)
    {
        cout << "Error: Unable to rename file" << endl;
        resetCurrentRecord();
    } else {
        cout << "File renamed successfully" << endl;
        AdminMenu();
    }

    // Declare variables to keep track of items sold
    int Ssingle = 0, Stwin = 0, Stowel = 0, Swater = 0, Sbiscuit = 0, Spillow = 0;
}

```

```

// Call the readCurrentRecord function to create a new current_record.txt file
readCurrentRecord();

}

void AdminMenu()
{
    system("cls");
    cout << "\t\t\t _____ Admin Menu _____ \n\n\n";
    cout << " ";
    cout << "\t| Press 1 to view room availability" << endl;
    cout << "\t| Press 2 to view amenities availability" << endl;
    cout << "\t| Press 3 to view sales report" << endl;
    cout << "\t| Press 4 to reset current record" << endl;
    cout << "\t| Press 5 to log out" << endl;
    cout << "\n\t\t\t Please enter your choice: ";
    cin >> choice; // Read the user's choice from the input
    cout << endl;

    switch (choice)
    {
        case 1:
            viewRooms(); // Call the viewRooms function to display room availability
            break;
        case 2:
            viewAmenities(); // Call the viewAmenities function to display amenities availability
            break;
        case 3:
            salesReport(); // Call the salesReport function to display the sales report
            break;
        case 4:
            resetCurrentRecord(); // Call the resetCurrentRecord function to reset the current record
            break;
        case 5:
            cout << "\t\t\t Logging out... \n\n"; // Display a logout message
            system("cls");
            main();
        default:
            cout << "\t\t\t Invalid choice. Please select from the options given above \n" << endl;
            AdminMenu(); // Call the AdminMenu function recursively
    }
}

void viewRooms()
{
    system("cls");
    // Display the number of available single and twin rooms
    int choice;
    cout << "\t\t\t _____ Room Availability _____ \n";
    cout << "\t\t\t Single Rooms: " << Qsingle << endl;
    cout << "\t\t\t Twin Rooms: " << Qtwin << endl;
    cout << "\n\n";

    cout << "Do you want to exit? \n1. Yes\n2. No" << endl << "Action: ";
    cin >> choice;
    if (choice == 1)
    {
        cout << "\t\t\t Logging out... \n\n"; // Display a logout message
    }
    else
    {
        system("cls");
        AdminMenu();
    }
}

void viewAmenities()
{
    system("cls");
    //Display number of available amenities
    cout << "\t\t\t _____ Amenities Availability _____ \n";
    cout << "\t\t\t Towel: " << 75 - Stowel << endl;
    cout << "\t\t\t Pillow: " << 75 - Spillow << endl;
    cout << "\t\t\t Water: " << 100 - Swater << endl;
    cout << "\t\t\t Biscuit: " << 100 - Sbiscuit << endl;
    cout << "\n\n";

    cout << "Do you want to exit? \n1. Yes\n2. No" << endl << "Action: ";
    cin >> choice;
    if (choice == 1)
    {
        cout << "\t\t\t Logging out... \n\n"; // Display a logout message
    }
}

```

```

    else
    {
        system("cls");
        AdminMenu();
    }
}

void viewPrices()
{
    system("cls");
    // Display the prices for each item
    cout << "\t\t\t Room Prices \n";
    cout << "\t\t\t Room Type \t Price \n\n";
    cout << "\t\t\t Single \t RM65 \n";
    cout << "\t\t\t Twin \t RM100 \n";
    cout << "\n";
}

void salesReport()
{
    system("cls");

    // Calculate the total sales for all items
    int Total_single = Ssingle * 65;
    int Total_twin = Stwin * 100;
    int Total_towel = Stowel * 0;
    int Total_water = Swater * 2;
    int Total_biscuit = Sbiscuit * 6;
    int Total_pillow = Spillow * 0;
    int total_sales = Total_single + Total_twin + Total_towel + Total_water + Total_biscuit + Total_pillow;

    // Display the sales report
    cout << "\t\t\t Sales Report \n";
    cout << "\t\t\t Item Quantity \t Total Sales \n\n";
    cout << "\t\t\t Single Rooms \t " << Ssingle << " \t\t RM" << Total_single << endl;
    cout << "\t\t\t Twin Rooms \t " << Stwin << " \t\t RM" << Total_twin << endl;
    cout << "\t\t\t Towels \t " << Stowel << " \t\t RM" << Total_towel << endl;
    cout << "\t\t\t Drinks \t " << Swater << " \t\t RM" << Total_water << endl;
    cout << "\t\t\t Snacks \t " << Sbiscuit << " \t\t RM" << Total_biscuit << endl;
    cout << "\t\t\t Pillows \t " << Spillow << " \t\t RM" << Total_pillow << endl;
    cout << "\n\t\t\t Grand Total: \t\t RM" << total_sales << endl;
    cout << "\n";

    cout << "Do you want to exit? \n1. Yes\n2. No" << endl << "Action: ";
    cin >> choice;
    if (choice == 1)
    {
        cout << "\t\t\t Logging out... \n\n"; // Display a logout message
    }
    else
    {
        system("cls");
        AdminMenu();
    }
}

```