

**COMPUTER SCIENCE DEPARTMENT**

**Object Oriented Programming**

**FINAL PROJECT:**

**SMART HOME STIMULATION**

**GROUP MEMBERS:**

FIZZA KHAN **(2312415)**

HAYYA SAYYAD ABBASI **(2312398)**

SIRAJ ULLAH **(2312411)**

Code:

#include <iostream>

#include <string> // For room names

using namespace std;

// Abstract Base Class (Abstraction)

class Device {

protected:

int id;

bool status; // On and Off status

static int deviceCount; // Static member to track the number of devices

const int energyUsage; // Constant for energy usage (in watts)

public:

// Constructor

Device(int id, int energyUsage) : id(id), energyUsage(energyUsage), status(false) {

deviceCount++; // Increment device count

}

// Virtual Destructor (Polymorphism)

virtual ~Device() {

deviceCount--; // Decrement device count

}

// Pure Virtual Functions

virtual void turnOn() = 0;

virtual void turnOff() = 0;

virtual void displayStatus() const = 0;

// Static Function to get device count

static int getDeviceCount() {

return deviceCount;

}

// Getter for energy usage

int getEnergyUsage() const {

return energyUsage;

}

int getId() const {

return id;

}

};

// Initialize static member

int Device::deviceCount = 0;

// Derived Class: Light (Inheritance)

class Light : public Device {

int brightness; // Brightness level

public:

// Constructor

Light(int id) : Device(id, 60), brightness(0) {} // 60W energy usage for light

// Override turnOn function

void turnOn() override {

status = true;

brightness = 100; // Default brightness

cout << "Light " << id << " is turned ON with brightness " << brightness << "%.\n";

}

// Override turnOff function

void turnOff() override {

status = false;

brightness = 0;

cout << "Light " << id << " is turned OFF.\n";

}

// Adjust brightness

void adjustBrightness(int level) {

if (status) {

brightness = level;

cout << "Light " << id << " brightness adjusted to " << brightness << "%.\n";

} else {

cout << "Light " << id << " is OFF. Cannot adjust brightness.\n";

}

}

// Display status

void displayStatus() const override {

cout << "Light " << id << ": " << (status ? "ON" : "OFF") << ", Brightness: " << brightness << "%\n";

}

};

// Derived Class: Thermostat (Inheritance)

class Thermostat : public Device {

int temperature;

public:

// Constructor

Thermostat(int id) : Device(id, 200), temperature(22) {} // 200W energy usage for thermostat

// Override turnOn function

void turnOn() override {

status = true;

cout << "Thermostat " << id << " is turned ON. Current temperature: " << temperature << "°C.\n";

}

// Override turnOff function

void turnOff() override {

status = false;

cout << "Thermostat " << id << " is turned OFF.\n";

}

// Set temperature

void setTemperature(int temp) {

if (status) {

temperature = temp;

cout << "Thermostat " << id << " temperature set to " << temperature << "°C.\n";

} else {

cout << "Thermostat " << id << " is OFF. Cannot set temperature.\n";

}

}

// Display status

void displayStatus() const override {

cout << "Thermostat " << id << ": " << (status ? "ON" : "OFF") << ", Temperature: " << temperature << "°C\n";

}

};

// Class: SmartHome

class SmartHome {

Device\* devices[10]; // Fixed-size array for devices

int deviceCount; // Number of devices added

public:

SmartHome() : deviceCount(0) {}

// Add a device

void addDevice(Device\* device) {

if (deviceCount < 10) {

devices[deviceCount++] = device;

cout << "Device added. Total devices: " << deviceCount << "\n";

} else {

cout << "Cannot add more devices. Limit reached.\n";

}

}

// Control a device

void controlDevice(int id) {

for (int i = 0; i < deviceCount; ++i) {

if (devices[i]->getId() == id) {

int action;

cout << "1. Turn On\n2. Turn Off\n3. Adjust Brightness (if Light)\n4. Set Temperature (if Thermostat)\nChoose action: ";

cin >> action;

if (Light\* light = dynamic\_cast<Light\*>(devices[i])) {

if (action == 1) light->turnOn();

else if (action == 2) light->turnOff();

else if (action == 3) {

int brightness;

cout << "Enter brightness level (0-100): ";

cin >> brightness;

light->adjustBrightness(brightness);

}

} else if (Thermostat\* thermostat = dynamic\_cast<Thermostat\*>(devices[i])) {

if (action == 1) thermostat->turnOn();

else if (action == 2) thermostat->turnOff();

else if (action == 4) {

int temperature;

cout << "Enter temperature (°C): ";

cin >> temperature;

thermostat->setTemperature(temperature);

}

} else {

cout << "Invalid action.\n";

}

return;

}

}

cout << "Device with ID " << id << " not found.\n";

}

// Display status of all devices

void displayStatus() const {

cout << "Device Status:\n";

for (int i = 0; i < deviceCount; ++i) {

devices[i]->displayStatus();

}

}

};

// Main Function

int main() {

SmartHome home;

cout << "\n";

cout << " WELCOME TO ~SHF~ SMART HOME STIMILATION SYSTEM! \n";

cout << "\n";

string pin;

cout << "Enter your Smart Home pin: ";

cin >> pin;

if (pin != "0909") {

cout << "Invalid pin. Access denied.\n";

return 0;

}

int choice;

do {

cout << "\n1. Add Device\n2. Control Device\n3. Display Status\n4. Exit\nEnter your choice: ";

cin >> choice;

switch (choice) {

case 1: {

int deviceType, deviceId;

cout << "Enter device type (1 for Light, 2 for Thermostat): ";

cin >> deviceType;

cout << "Enter device ID: ";

cin >> deviceId;

if (deviceType == 1) {

home.addDevice(new Light(deviceId));

} else if (deviceType == 2) {

home.addDevice(new Thermostat(deviceId));

} else {

cout << "Invalid device type.\n";

}

break;

}

case 2: {

int deviceId;

cout << "Enter the ID of the device to control: ";

cin >> deviceId;

home.controlDevice(deviceId);

break;

}

case 3:

home.displayStatus();

break;

case 4:

cout << "Exiting...\n";

break;

default:

cout << "Invalid choice.\n";

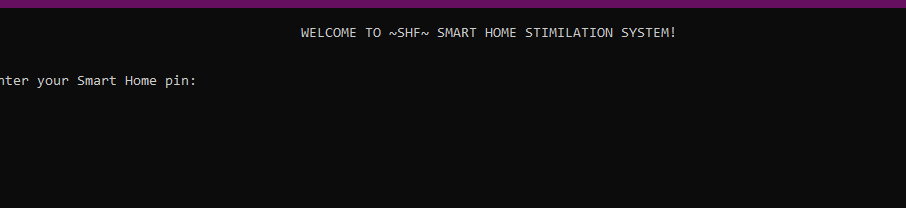
}

} while (choice != 4);

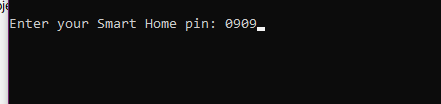
return 0;

}

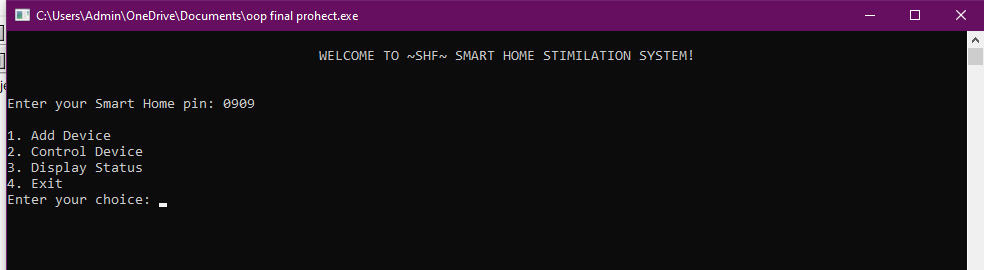
OUTPUTS:

**1.MAIN SCREEN**

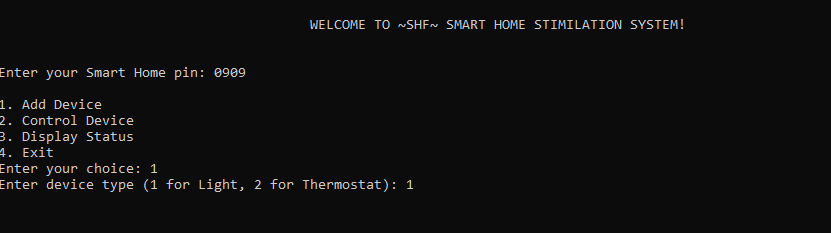
**2 PIN OF OUR SMART HOME**



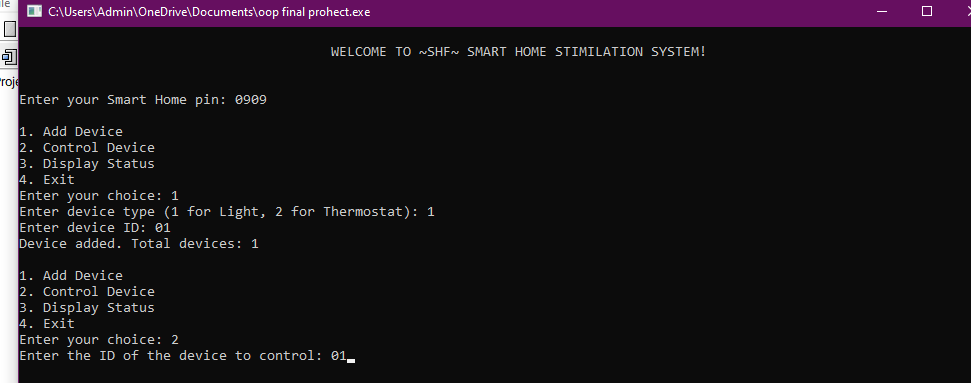
**3 IF PIN IS COREECT THEN ACCESS IS GRANTED TO LIGHTS AND TEMPERATURE MANAGEMENT**



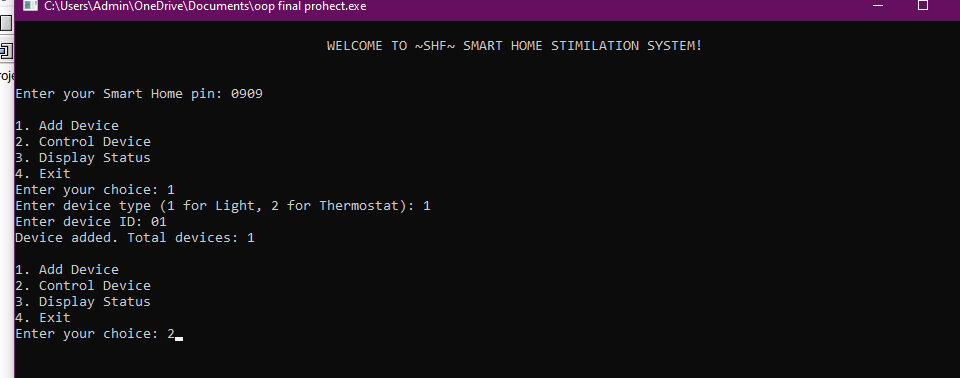
**4 FIRST INPUT IS OF LIGHT**



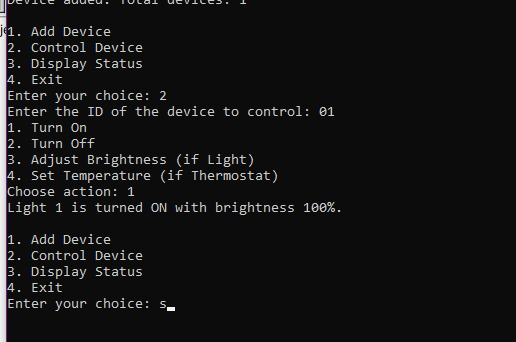
**5 ID OF LIGHT IS “01”**



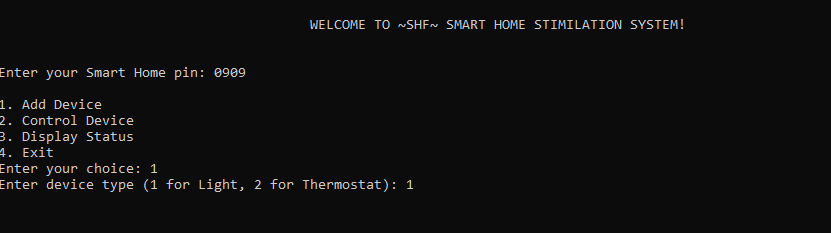
**6 THE LIGHT WITH ID “ 01” IS TURNED ON FIRST SO THAT ITS BRIGHTNESS CAN BE CHANGED**



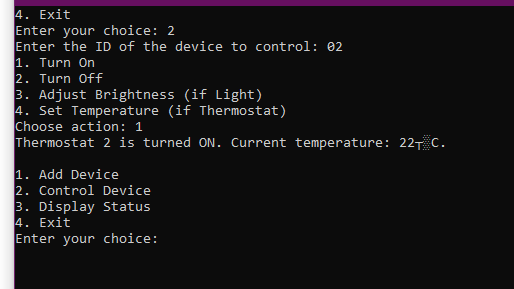
**7 THE BRIGHTNESS OF LIGHT WITH ID “01’ IS SET TO 100**



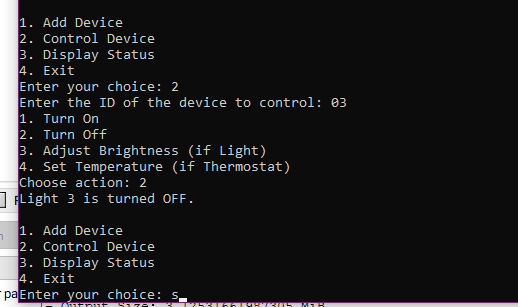
**8 NOW THERMOSTAT**



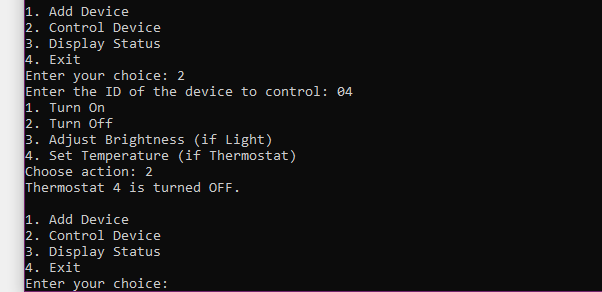
**9 ID OF THERMOSTAT IS “02” AND ITS TEMPERATURE IS SET TO 22 C**



**10 NOW ANOTHER LIGHT WITH ID “03” IS ADDED AND IT IS TURNED OFF**

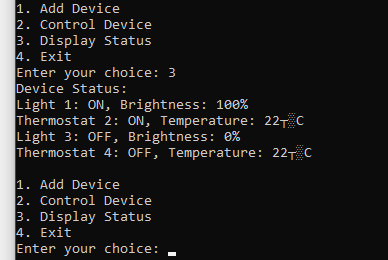


**11 ANOTHER THERMOSTAT WITH ID “04 “ IS ADDED AND IT IS TURNED OFF TOO**



12

**12 FOLLOWING IS THE NUMBER OF DEVICES THAT ARE TURNED ON AND OFF IN OUR HOME**



**13 EXITING THE SYSTEM**

