

United International University

CSE 1116 - Coding Test 2 - Set A

Write your name and ID on the question paper. Marks are on the right. Return question.

Q1. Smart Home Device System (20 Marks)

A company wants to simulate different smart home devices using object-oriented design. You need to design classes using **abstract classes and interfaces**.

Requirements

1. Interface: Connectable [5]

- Method: void connectToWiFi(String network)
- Method: void disconnectWiFi()

2. Abstract Class: SmartDevice [5]

- Fields: String name, boolean powerStatus
- Constructor: (String name) : sets powerStatus to false by default
- Methods:
 - void turnOn() : sets powerStatus to true
 - void turnOff() : sets powerStatus to false
 - abstract void showStatus()

3. Class: SmartLight - subclass of SmartDevice and uses Connectable [5]

- Additional field: int brightness
- Implement Wi-Fi methods with proper messages (e.g. Connected to the given network/ Disconnected etc.)
- Override showStatus() to display name, power status, and brightness
- Add void adjustBrightness(int value) : if value out of range (0–100), print “Invalid brightness level.”

4. Main Class: HomeAutomationSystem [5]

- Create 2 objects of SmartLight
- Demonstrate:
 - Turning them on/off
 - Connecting/disconnecting Wi-Fi
 - Adjusting brightness
 - Calling showStatus() after each operation to show the changes

Q2. Bank Transaction Handling (10 Marks)

A bank application processes money transfers between accounts. Implement exception handling to make the system robust.

Requirements

1. **Class: InsufficientBalanceException** (user-defined, checked) [2]
 - Subclass of Exception
 - Constructor: takes message
2. **Class: Account [5]**
 - Fields: String accNumber, double balance
 - Constructor: (String accNumber, double balance)
 - Method:
 - void deposit(double amount) : if amount ≤ 0 , throw IllegalArgumentException
 - void withdraw(double amount) : if amount > balance, throw InsufficientBalanceException
3. **Main Class: BankApp [3]**
 - Create two Account objects
 - Try depositing invalid amount : catch and print message
 - Try withdrawing more than balance : catch user-defined exception
 - Use finally block to print “Transaction process completed.” for both above operations
 - Show valid deposit/withdraw case too