Department of Statistics & Computer Science
University of Kelaniya
ACADEMIC YEAR - 2022/2023
BECS 12243 / BECS 12243(R)
Object Oriented Programming

Practical Examination

Date: 20/01/2025 **Duration: 2 hours**

Instructions (Read carefully)

- Create a folder in the Desktop and name it as YourLastName-2022-XXX, where XXX is the last 3 digits of your student number. (This folder is your working directory)
- Create an IntelliJ Java project inside above folder with the name *YourLastName-2022-XXX*.
- You are not allowed to access the Internet or access any other files and folders other than your working directory.
- Make sure you do not keep mobile phones or any electronic device with you during the exam.

- **Submission:**

  You should submit the following two items into the answer submission folder in EVAL,

  ✓ Full Answer as a MS word document:
    - Include the Content of the Java files (complete source code) into a single word document. Name this document with your student number EC22-XXX. (the header of this document should be your student ID, name, and the course code)
    - Include the screenshot of the final output of the program into above word document.

  ✓ IntelliJ project as a zipped file.

- Note: You are responsible to save your work always including the word file and submit above two items. Once you submit, verify your submission again.

A Smart Home Device Management System is designed to manage various types of smart devices, including lights, thermostats, and security cameras. Each device type has a method called ***calculateEnergyConsumption*** that calculates the **monthly energy consumption** based on device-specific attributes. For lights, the calculation depends on the wattage and daily usage hours. For thermostats, it considers the average runtime hours per day and power rating. For security cameras, it factors in the resolution (in megapixels) and the hours of operation per day.

Only thermostats and security cameras have the ability to optimize their energy usage dynamically based on a specified percentage adjustment in runtime hours or resolution.

The required interfaces and classes for this system are outlined below.

Interfaces and Classes:

- ***SmartDevice*** interface with a method ***calculateEnergyConsumption***.
- ***OptimizableEnergy*** interface with a method ***optimizeEnergyUsage***.
- ***Light*** class implementing ***SmartDevice.***
- ***Thermostat*** and ***SecurityCamera*** classes implementing both ***SmartDevice*** and ***OptimizableEnergy.***

Specifications:

The energy consumption rates for different devices are as follows:

Light: Energy (kWh) = (Wattage × Daily Usage Hours × 30) / 1000;
Thermostat: Energy (kWh) = (Average Runtime Hours × Power Rating × 30) / 1000;
Security Camera: Energy (kWh) = (Resolution × Hours of Operation × 0.05 × 30).

Note: The 0.05 factor represents the typical energy usage per megapixel-hour.

Part I: Write the code for all the interfaces and classes to complete the Smart Home Device Management System. Each class should include:

- A constructor initializes attributes, including the device's name.
- A toString method to provide a string representation of the device that includes its name, and current energy consumption.

(Hint: Consider the device objects given in Part II.)

[50 Marks]

Part II: Create a SmartHomeManager class to manage the smart devices and perform the following tasks: [30 Points]

1. Create the following devices using their respective constructors:

   ```
   SmartDevice livingRoomLight = new Light("Living Room Light", 60, 5);
   SmartDevice bedroomThermostat = new Thermostat("Bedroom Thermostat", 8, 1.5);
   // For a Thermostat with a power rating 1.5
   SmartDevice outdoorCamera = new SecurityCamera("Outdoor Camera", 4, 12);
   // For a 4 MP camera
   ```

2. Display the details of each device including monthly energy consumption using their toString method.

Part III: As energy optimization is crucial for smart home devices, perform the following tasks:

[20 Marks]

1. Reduce the runtime hours of the bedroomThermostat by 15% using the optimizeEnergyUsage method and display the updated details using toString.

2. Adjust the resolution of the outdoorCamera by reducing it by 10% and display the updated details using toString.

(Hint: Use polymorphism and the optimizeEnergyUsage method where applicable.)

----------End of the Paper ----------