School of Computing Technologies

COSC1295 Advanced Programming 2025 S1

Assessment 1

⌘	Assessment Type: **Individual assignment; no group work.** Submit online via Canvas → Assignments → Assignment 1. Marks awarded for meeting requirements as closely as possible. Clarifications/updates may be made via announcements/relevant discussion forums.
📅	Due date: **Week 7**; Deadlines will not be advanced but they may be extended. Please check Canvas → Assignments → Assignment 1 for the most up to date information. As this is a major assignment, a university standard **late penalty** of 10% per each working day applies for up to 5 working days late, unless special consideration has been granted.
✓	Weighting: **18 marks** out of 100

Background Information

For this assessment you need to write an object-oriented console application in the Java programming language which adheres to the following basic object-oriented programming principles:

a) Your code should follow good object-oriented principles such as encapsulation, composition, inheritance, etc…
b) Setting the visibility of all instance variables to private or protected.
c) Using getter and setter methods only where needed and appropriate with consideration given to scope & visibility.
d) Only use static variables and methods when there is good reason to do so, such as reducing code repetition whilst still maintaining good objected-oriented design.
e) Using superclass methods to retrieve and / or manipulate superclass properties from within subclasses, avoiding direct access to class attributes if possible.
f) Any collections of objects in your program should be stored properly.

You are being assessed on your ability to write a program in an object-oriented manner. Writing the program in a procedural style is not permitted and will be considered inadmissible receiving a **zero** grade.

1. Overview

You are to implement a smart home automation system that manages different electronic devices in a house. The system allows users to control various appliances such as Lights and Fans. The system will be able to turn a smart home device ON/OFF and visualize the status of all devices. You will be implementing a series of classes to meet these needs. The classes must be designed according to object-oriented principles.

2. Assessment Criteria

This assessment will determine your ability to:

1. Follow coding, convention and behavioral requirements in this specification and in the class.
2. Independently solve a problem by using concepts taught over the first several weeks.
3. Write and debug Java code independently.
4. Document code.
5. Provide references where due.
6. Manage the code development process and meet deadlines.
7. Seek clarification from your “client”/“supervisor” (us) when needed via discussion forums.
8. Create a program by recalling concepts taught in class, understanding and applying concepts relevant to solution, analysing components of the problem, evaluating different approaches.

3. Learning Outcomes

1. Demonstrate knowledge of basic OO concepts and structure design in Java programming.
2. Devise solutions to computing problems under specific requirements.
3. Encode the devised solutions into executable programs and test the programs.
4. Demonstrate understanding of coding conventions and ethical considerations in programming.
5. Manage time properly.

4. Getting Started

The Smart Home Automation System – Stage 1

In Stage 1, we assume a system with 2 smart home devices by default: a Light and a Fan. When launching the program, both devices are OFF. The system will be able to:

- Show all devices.
- Turn ON device
- Turn OFF device
- Adjust Device
- Exit

4.1 Define an Interface `SmartDevice`

All smart devices to be managed must implement a common interface that allows users to:

- Turn ON the device
- Turn OFF the device
- Show device status

4.2 Since all appliances share common properties (e.g., name and power status), define a parent class `Appliance` that implements the `SmartDevice` interface

- An appliance has two properties: name and power status
- An appliance can turn ON
- An appliance can turn OFF
- An appliance can show its own status (ON or OFF)

4.3 Create subclass `Light` that extends the `Appliance` class

- A Light has an extra property: brightness (int). Brightness can only be within [1, 10]
- A Light can adjust its own brightness
- When showing status, a Light can show if it is ON or OFF. If ON, a Light will show its own brightness
- When a Light is turned ON, its brightness is set as 5 by default

4.4 Create subclass `Fan` that extends the `Appliance` class

- A Fan has an extra property: speed (int). Speed can only be within [1, 5]
- A Fan can set its own speed
- When showing status, a Fan can show if it is ON or OFF. If ON, a Fan will show its speed
- When a Fan is turned ON, its speed is set as 1 by default

4.5 Implement a Main class to drive the program

The Main class implements a menu for users to navigate different functionalities of the program:

- Show All Devices – Show and visualize the status of all devices
- Turn ON Device – Prompt the user to select a device to turn on
- Turn OFF Device – Prompt the user to select a device to turn off
- Adjust Device – Prompt the user to select a device to adjust (e.g., set fan speed or light brightness)
- Exit – Quit the program

Throughout this assignment, we assume that:

- User will always enter a valid input. That is, input validation is not required.
- Keyword matching here is **case insensitive**, for example, when prompting user to enter "y/n" for confirmation, "Y" is equivalent to "y".

Following is an example of interactions with the automation system. You do not have to strictly follow the format but it illustrates required functionality. Text in bold and green is input from the user.

```
===============================================
Welcome to Smart Home Automation System!
===============================================
Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Exit
Please select: 1
*** Showing All Devices ***
Light: OFF
Fan: OFF


Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
```

```
5. Exit
Please select: 2
Which device do you want to turn ON: Light
Light is now ON.

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Exit
Please select: 2
Which device do you want to turn ON: Light
Light is already ON.

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Exit
Please select: 1
*** Showing All Devices ***
Light: ON
[█████-----] 5/10
Fan: OFF


Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Exit
Please select: 4
Which device do you want to adjust: Light
Enter brightness level [1, 10]: 9
Light brightness set to 9.


Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Exit
Please select: 4
Which device do you want to adjust: Fan
Fan is OFF. Do you want to turn on FAN (y/n): Y
Fan is now ON.
Enter speed level [1, 5]: 1
Fan speed set to 1.


Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Exit
Please select: 3
Which device do you want to turn OFF: Light
Light is now OFF.
```

```
Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Exit
Please select: 3
Which device do you want to turn OFF: Light
Light is already OFF.

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Exit
Please select: 1
*** Showing All Devices ***
Light: OFF
Fan: ON
[█----] 1/5

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Exit
Please select: 5
Goodbye!
```

The Smart Home Automation System – Stage 2

In this stage, we enhance the system by allowing user to add more smart home devices to the system. The user can add lights or fans at different locations of the home, for example, living room light, study room fan, etc.

4.6 Use appropriate JCF data structure to store all smart home devices

- When first launching, the system does not have any smart home devices
- User can add a smart home device (either a Light or a Fan) to the system
- User can remove an existing device from the system
- Any device added to the system should have a unique device name – duplicated names should not be allowed

Following is an example of interactions with the automation system. You do not have to strictly follow the format but it illustrates required functionality. Text in bold and green is input from the user.

```
===============================================

Welcome to Smart Home Automation System!

===============================================

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
```

```
Please select: 1
*** Showing All Devices ***


Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
Please select: 2
You do not have any devices now.

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
Please select: 5
Enter device type (Light/Fan): Light
Enter device name: Living Room Light

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
Please select: 1
*** Showing All Devices ***
Living Room Light: OFF

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
Please select: 5
Enter device type (Light/Fan): Light
Enter device name: living room light
The device cannot be added because it already exists in the system.

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
Please select: 5
```

```
Enter device type (Light/Fan): Fan
Enter device name: Study Fan

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
Please select: 1
*** Showing All Devices ***
Living Room Light: OFF
Study Fan: OFF

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
Please select: 2
Which device do you want to turn ON: Study Light
Sorry! This device does not exist.

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
Please select: 2
Which device do you want to turn ON: living ROOM LIGHT
Living Room Light is now ON.

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
Please select: 1
*** Showing All Devices ***
Living Room Light: ON
[█████-----] 5/10
Study Fan: OFF

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
```

```
Please select: 6
Which device do you want to remove: fan
Sorry! This device does not exist in the system.

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
Please select: 6
Which device do you want to remove: STUDY fan
Study Fan has been removed from the system.

Choose an option:
1. Show All Devices
2. Turn ON Device
3. Turn OFF Device
4. Adjust Device
5. Add a Device
6. Remove a Device
7. Exit
Please select: 1
*** Showing All Devices ***
Living Room Light: ON
[█████-----] 5/10
```

Marking Guide

This assessment is divided up into several components:

• Basics (OO design, menu, show all devices)	(5 marks)
• Adjusting device (Turn ON/OFF, set brightness/fan speed)	(3 marks)
• Adding device	(4 marks)
• Removing device	(4 marks)
• Code quality (code readability, comment)	(2 marks)

Submission Instructions

1. Submit a zip file

Export the project as an archive; the file name should be the same as the project name, for example **s1234567-APAssignment1.zip** and submit the zip file to Canvas under Assignments.

To export a project as a zip file in Eclipse/IntelliJ, click **File** -> **Export** -> **Project to zip file**.

NOTE: you need to make sure your code compiles and runs properly on command line.

You are required to submit regularly, at least **one submission per week** unless you would like to complete early. Canvas will keep all submissions. **No extension request will be granted** if there is no sufficient amount of submissions.

Marking is purely based on the last version. Early submissions will not be assessed.

2. Submit a demonstration video

A video demonstration of your code. In the video, please explain the class design of your program with a small demonstration of the functionality of your program. The video should be no longer than 10 minutes.

For the video demo, use Microsoft Streams for creating and sharing your recording. MS Streams is very easy to use. Please share your video demo with all teaching staff. Please see Canvas -> Assignments -> Assignment 1 for detailed instructions. You must show your face while you demo. Please leave a link to your Microsoft Streams Video as a comment to your submission in Canvas.

Please do NOT send email reminders when sharing your video in MS Streams.

Academic Integrity and Plagiarism (Standard RMIT Warning)

Your code will be automatically checked for similarity against other submissions so please make sure your submitted project is entirely your own work.

Academic integrity is about honest presentation of your academic work. It means acknowledging the work of others while developing your own insights, knowledge, and ideas. You should take extreme care that you have:

- Acknowledged words, data, diagrams, models, frameworks and / or ideas of others you have quoted (i.e., directly copied), summarised, paraphrased, discussed, or mentioned in your assessment through the appropriate referencing methods.
- Provided a reference list of the publication details so your reader can locate the source if necessary. This includes material taken from internet sites.

If you do not acknowledge the sources of your material, you may be accused of plagiarism because you have passed off the work and ideas of another person without appropriate referencing, as if they were your own.

RMIT University treats plagiarism as a very serious offence constituting misconduct. Plagiarism covers a variety of inappropriate behaviours, including:

- Failure to properly document a source.
- Copyright material from the internet or databases.
- Collusion between students.

For further information on our policies and procedures, please refer to the University website.

Assessment Declaration

When you submit your project electronically, you agree to the RMIT Assessment Declaration.