

**CHRIST (Deemed to be University)**  
**Department of Computer Science**  
**Master of Artificial Intelligence and Machine Learning**

**Course:** MAI271 – JAVA Programming

**Exercise No:** LAB Exercise – 4

**Date:** 12 – 11 – 2024

**Duration:** 2 Hrs

Establish an abstract class named Robber with a function named RobbingClass to print “MScAI&ML”. Create a class named JAVAProfessionalRobber that inherits from Robber and implements four abstract methods: RowHouses(), RoundHouses(), SquareHouse(), and RectangleHouse(). Additionally, include a default method called MachineLearning in the abstract class, which prints "I love MachineLearning." **(2 Marks)**

The constraints for the four abstract methods are outlined as follows:

- **RowHouses():** Accepts an integer array representing the amount of money in each rowhouse. The objective is to maximize the amount robbed without triggering security alarms from adjacent houses. The method should return the maximum amount that can be robbed without getting caught. **(2 Marks)**
- **RoundHouses():** Accepts an integer array representing the amount of money in each roundhouse. Since the houses form a circle, with the last and first houses being adjacent, the goal is to maximize the amount robbed without triggering security alarms from adjacent houses. The method should return the maximum amount that can be robbed without getting caught. **(2 Marks)**
- **SquareHouse():** Accepts an integer array representing the amount of money in each square house. Robbing adjacent square houses triggers security alarms. The method should return the maximum amount that can be robbed without getting caught. **(2 Marks)**
- **MuliHouseBuilding():** Accepts an integer array representing the amount of money in each type of house in a multi-type building. Robbing adjacent houses of any type triggers security alarms. The method should return the maximum amount that can be robbed without getting caught. **(2 Marks)**

**Test Cases:**

- RowHouses([1,2,3,0]) -> 4
- RoundHouses([1,2,3,4]) -> 6
- SquareHouse([5,10,2,7]) -> 17
- MultiHouseBuilding([5,3,8,2],[10,12,7,6],[4,9,11,5],[8,6,3,7]) -> 59

**General Instruction:**

1. Ensure that your code includes relevant comments to enhance readability and understanding. Subsequently, upload your code to GitHub for version control and collaborative access.
2. Include descriptive comments within the code, explaining its functionality and logic.

3. In the Google Classroom submission, include the GitHub URL where your code is hosted.
4. Attach a PDF document named "your\_register\_number\_exercise\_No.pdf" to the submission. The PDF document should include screenshots of the code and the output screen.
5. Upload the answer document&GitHub URL in Google Classroom on or before the deadline mentioned. Evaluation will not be considered for late submission

**Evaluation Rubrics:**

**Declaration of abstract class & methods:**2 Marks

**Execution of Method 1:**2 Marks

**Execution of Method 2:**2 Marks

**Execution of Method 3:**2 Marks

**Execution of Method 4:**2 Marks

**Total:**10 Marks