**CMPU 2016 Object Oriented Programming**

TU857-2

2024-25, Semester 1: Python with Sunder Ali Khowaja

[SunderAli.Khowaja@tudublin.ie](mailto:SunderAli.Khowaja@tudublin.ie)

**Lab 6: Working with Python Objects - Magic Methods and Operator Overloading**

Welcome to Week 6 of the Object-Oriented Programming course! In this lab, you will dive into the world of magic methods and operator overloading in Python. These advanced concepts allow you to define custom behaviour for built-in operations and interactions with your classes. You'll build upon the existing game code from previous weeks and enhance it by incorporating these powerful techniques.

**Lab Duration**: 2 hours.

**Lab Objectives:**

* Understand the concept of magic methods and their role in Python classes.
* Learn to use magic methods to customize the behaviour of built-in operators.
* Explore the concept of operator overloading to provide meaningful interactions with your classes.
* Apply magic methods and operator overloading to the existing game code to enhance its functionality.

**Instructions:**

1. Open the Python code file from the previous week's lab (Week 5). You can choose to use your own code or begin from the uploaded solution.
2. Review the existing codebase, including the Character, APC, Suspect, Witness, and Game classes.
3. Implement magic methods (e.g., \_\_str\_\_, \_\_repr\_\_, \_\_eq\_\_, \_\_lt\_\_, \_\_add\_\_):
   1. Extend the Character class to be able to print derived classes with their class name and the \_name variable of each instance.
   2. Extend the Character class to overload the < and the == operators. Use the \_name variable to define behaviour.
   3. Extend the Suspect class to provide a formal representation using \_\_repr\_\_, which should output the \_name, \_dialogue and \_alibi content of the instance’s variables.
   4. Extend the Witness class to overload the + operator. Use the \_observation variable to define behaviour. The operator should return a new Witness object containing the concatenated observations.
4. In the section “if \_\_name\_\_ == "\_\_main\_\_":” comment out all the lines related to creating a Game. Today, we won’t play the full game but just focus on the behaviour of the magic methods and our operator overloading (see below for example outputs):
   1. Create 2 Suspect and 2 Witness instances.
   2. Let them interact().
   3. print(…) all instances.
   4. Compare two of the instances using the == operator.
   5. Compare the names of two of the instances using the < operator.
   6. Create a new Witness by combining your two witnesses using the + operator. print(…) the new Witness. share\_observation() for the new Witness.
5. Test the enhanced classes by utilizing the newly defined magic methods and overloaded operators.
6. Challenge Task (optional – bonus points section):
   1. Include the new operator overloading into proper game play.
   2. Create a custom magic method that calculates and returns a score based on interactions with characters in the game. Use this score to evaluate the player's performance at the end of the game.
7. Answer the quiz questions and upload your code. This lab is marked. A solution is available from tomorrow on. The solution code will be discussed in the next lecture, which takes place after review week (the review week exercise will also be discussed in that lecture). Your solution will likely differ from mine. If you fulfil the task description this is not a problem and is to be expected. If you have questions after reviewing the solution please contact me asap via email.

**Example output from the “if \_\_name\_\_ == "\_\_main\_\_":” section:**

Alice: I was at home.

Bob: I was at the park.

Carol: I saw someone running away.

Dave: I heard a loud noise.

Magic Methods and Operator Overloading Example:

Suspect: Alice

Are Alice and Bob the same person? False

Are Bob and Bob the same person? True

Is Bob before Carol? True

Who is the new Witness? Witness: Carol and Dave

Carol and Dave's Observation: They had a hat. and The noise came from the East.