# Evolution through Programming

Lecturer: Prof. Yitzhak Pilpel  
TA: Omer Kerner – [omer.kerner@weizmann.ac.il](mailto:omer.kerner@weizmann.ac.il)

## Assignment 3 - Evolution of Cooperation

**Objective:**

Your mission, should you choose to accept it, is to devise a clever and robust Python-based strategy for the repeated Prisoner's Dilemma. Your strategy will enter a digital arena where it will battle head-to-head against the scripts created by your fellow classmates. This is your chance to demonstrate strategic thinking, coding skills, and creativity.

The ultimate champion strategy will earn fame and glory!

**Rules and Guidelines:**

1. **Python Compatibility:**
   * Your strategy must be implemented as a Python script.
   * It should define a function named strategy() which takes two arguments:
     + my\_history: a list containing your past moves.
     + opponent\_history: a list containing your opponent’s past moves.
   * The function must return either 'cooperate' or 'defect'.
   * **Example:**

def strategy(my\_history, opponent\_history):

    # Simple example: Always cooperate

    return 'cooperate'

1. **Repeated Interaction:**
   * Your strategy will play repeatedly against each opponent's strategy.
   * Your script should remember past moves and adapt accordingly.
2. **Fair Play:**
   * Avoid using external libraries or storing data externally. The script should be self-contained.
   * Your code must execute quickly—no more than 1 second per round.
3. **Documentation and Strategy Explanation:**
   * Alongside your script, submit a short explanation (1-2 paragraphs) detailing your strategic logic.
   * Explain why you chose your strategy and how you expect it to perform against different types of opponents (e.g., always defect, tit-for-tat, random, etc.).

**Evaluation Criteria:**

* Effectiveness: How well your strategy performs against a variety of opponents.
* Adaptability: Your strategy’s ability to adjust based on the opponent’s moves.
* Originality and creativity: Unique strategies will stand out!
* Clarity of explanation: Clearly articulated strategic thinking.

**Submission:**

Submit a single Python script file (yourname\_strategy.py) containing your strategy function.

Include your strategic explanation at the beginning of your script as comments.