**Project: Develop a Chess Game in Python**

**Objective**

Students will implement a simplified version of the classic chess game using Python. This project aims to reinforce concepts of object-oriented programming, algorithms, and game logic.

**Requirements**

1. **Basic Structure:**
   * Create a Board class to represent the chessboard.
   * Create a Piece class as a base class for all chess pieces (Pawn, Rook, Knight, Bishop, Queen, King).
   * Each piece class should inherit from Piece and implement its own movement logic.
2. **Game Logic:**
   * Implement methods to initialize the board with the standard chess setup.
   * Allow players to make moves by specifying the piece and the destination square.
   * Validate moves according to chess rules (e.g., a pawn can only move forward, bishops move diagonally, etc.).
   * Implement basic rules (e.g., check and checkmate) but avoid advanced rules (like castling, en passant, or pawn promotion) for simplicity.
3. **User Interface:**
   * Provide a simple text-based interface for players to input their moves (e.g., e2 to e4).
   * Print the current state of the board after each move.
   * Handle invalid moves gracefully by informing the user and prompting for another input.
4. **Game Loop:**
   * Implement a loop that alternates turns between two players until the game ends (either checkmate or stalemate).
5. **Optional Features (Bonus):**
   * Implement a simple AI opponent using random moves or basic strategies.
   * Add support for saving and loading games.
   * Enhance the user interface (e.g., using a graphical library like Pygame).

**Deliverables**

* A Python script containing the complete code for the chess game.
* A README file explaining how to run the program and any assumptions made during development.
* Comments in the code to clarify the logic and structure.

**Evaluation Criteria**

* **Code Quality:** A README file explaining how to run the program and any assumptions made during development
* **Functionality:** The game should run without errors and follow the basic rules of chess.
* **User Experience:** The interface should be intuitive, and instructions should be clear.
* **Creativity:** Any additional features or enhancements beyond the basic requirements.

**Submission Guidelines**

* Submit your code as a .zip file to [mehmetkutlay.kocer@ue-germany.de](mailto:mehmetkutlay.kocer@ue-germany.de) by 11.01.2025.  
  ClassName\_TeamName\_Member1Name\_Member2Name.zip
* Ensure that your code is well-documented and that you include your name and student ID in the README file.

**Resources**

* Python documentation: [Python.org](https://docs.python.org/3/)
* Chess rules: chess.com