ASSIGNMENT INSTRUCTIONS

Course: Basic Programming 2
Project Topic: Simulation of an object, event, or scenario from a film or game, implemented in the backend

1. Project Description

Students are expected to choose an object (e.g., Iron Man suit), event, or scenario from a movie or game and develop a command-line or script-based simulation. The simulation must incorporate all topics covered during the course (OOP, functional programming, I/O, error handling, etc.).

2. Deliverables

- **Code:** A fully functional backend application (no frontend required).
- **Report:** A short written document summarizing the selected topic, system design, and techniques used.

3. Presentation and Submission Schedule

Stage	**Date**
1. First Presentation (MVP Demo)	14.05.2025
2. Second Presentation (Refined)	21.05.2025
3. Final Submission (Code + Report)	28.05.2025

- After each presentation, questions will be asked in class. Each group member must answer at least one question.
- Groups will consist of 1–3 students.
- The complexity of the selected topic will positively or negatively affect the project grade.

4. Evaluation Criteria

1. **Functionality:** Scope and correctness of the simulation
2. **Code Quality:** Modularity, readability, and use of comments

3. **Use of Course Topics:** The extent to which the checklist items below are implemented
4. **Presentation & Q&A:** Clarity of presentation and responses to questions
5. **Report:** Clear, concise, and consistent documentation

5. Course Topics Checklist

- **Functional Programming:** Are structures like lambda and map/filter/reduce used for data processing or simulation logic?
- **Modules & Package Management:** Is the code organized into logical modules? Are custom modules used?
- **File Handling & Persistence:** Are txt/csv/json formats used for reading/writing data? (e.g., saving simulation history)
- **Error Handling:** Are try/except blocks used properly? Are custom exceptions defined where necessary?
- **OOP (Classes, Objects):** Are the main structures modeled using classes? Is __init__, method use, and self applied correctly?
- **Inheritance & Polymorphism:** Are there subclasses with overridden methods? (e.g., different modes or event types)
- **Time & Utility Modules:** Are Python modules such as datetime, random, re, itertools, collections used meaningfully?

Note: A frontend is not required. The simulation's entire logic and user interaction must be handled through the terminal.

Note: Students must send the names of their group members and their selected topic via email to the instructor by **10.05.2025**.

E-mail: timucin.utkan@istinye.edu.tr