Specify a project:

(i) The student-course Assignment problem:

Input: Students, each Student's preference for each course or general student priority
Output: Student assignment to, *optimize the total satisfaction* or *maximize the minimum satisfaction*.

- How many courses should each student take?
- Use object oriented programming.

Description of the project:

The problem of assigning students to courses is a classic combinatorial optimization problem that arises in many real-world applications. The problem consists of assigning students to courses based on their preferences in a way that *maximizes total satisfaction* or *maximizes the minimum satisfaction*.

Steps to be followed:

Data collection: Collect data on students and their course preferences. Data can be collected through surveys or questionnaires, but in this case, the author will try to find a dataset in Kaggle and if there is nothing encountered, he will proceed to make up the data with random values.

Model formulation: Formulate the problem of assigning courses to students as an optimization problem. There are different approaches to model this problem, such as using a linear programming formulation or a combinatorial optimization formulation.

Algorithm design: Design an algorithm to solve the optimization problem. For instance, greedy algorithms, genetic algorithms, integer linear programming, simulated annealing or Tabu Search, to solve the problem.

Implementation: Implement the algorithm using a programming language of your choice. In this case, *Python* will be used.

Testing and validation: Test the algorithm using different scenarios and data sets to ensure that it produces optimal or near-optimal solutions.For instance, adding parameters to the model such as the maximum number of courses that a student can take, or a maximum number of students that a course can have.

Performance evaluation: Evaluate the performance of the algorithm by comparing it to other existing algorithms or heuristics on the same problem case.

Analysis and interpretation of results: Analyze and interpret the results of the algorithm to better understand the problem and its solution. You can compare the results with learner preferences and identify patterns or trends. Also check if the results are definitely optimal or not.