CodeJam() 2016 Junior Challenge

Problem Background

One of the prevalent problems in student life is the coordination of student schedules. Many schools remedy this problem by operating on a 'first' to register, first to enter basis. As fair as this method is, it often results in student programs being delayed due to inability to get into classes. It also results in long periods of downtime of the registration website due to the frenzy of students trying to register for courses on time. A new solution is required.

The Challenge

HCI College has 80 students to enroll in the fall. 10 classes will be offered; each class can receive a maximum of 20 students. For all students, each class is equal in terms of credits and prerequisites. The same class is given twice a week by the same teacher and a student can pick only one of the 2 scheduled choices.

Each student should be taking 5 classes this semester.

The class schedule looks like this (complete schedule of classes in the INPUT - in the following):

Class ID	**Class Name**	**Class Time**
101	Mathematics	Monday: 8:30am-10:30am, Tuesday: 8:00am-10:00am
102	Physics	Tuesday: 8:30am-10:30am, Wednesday: 8:00am-10:00am
103	Biology	Wednesday: 8:30am-10:30am, Friday: 8:00am-10:00am
...	...	...

This semester the schedule needs to be put together efficiently and quickly. HCI College is looking to a group of intelligent students from another school (your team) to design a system that schedules each student based on a single input parameters - each student's availability for each working day of the week (Monday: 8:30am - 5:30pm, Tuesday: 8:00am-2:30pm, Thursday: 8:30am-5:30pm, Friday: 9:30am-1:30pm).

Note:

• Notice that some students are not available on some days (i.e. Saturday, Sunday and some week days – Availability: 0h)

There are several schedules that can be generated. Your program should find the **BEST** scheduling of all students. (Hint: This is an optimization problem). An ideal solution should fill all classes to the maximum class size of 20 students/class.

Givens

You will be given the following:

- JSON file that contains the Classes data (classes.json) and student data input for Availability (studentsWithAvailability.json)
- Example of how your solutions should be formatted in the JSON output file (codejam-challenge-example.json)

Note: The environment where you write and run your program is yours to pick, but you need to give us a way to access it in order to evaluate your solution (provide us a READ-ME.txt file).

Deliverables

To be considered for judging the implemented solution must meet the base requirements. Winners will be chosen from solutions, eligible for judging, based on success of code and demonstration of creativity when meeting the themed requirement. Although this problem is one part computational and one part interface design, the interface design portion carries more weight in choosing winners.

- **Base Requirement:**
  - Optimal solution output file (result of your program execution) in JSON file format titled <codejam-challenge.json>. Structure of this file should be same as the structure in <codejam-challenge-example.json>.

- **Themed Requirement**
  - An interactive output interface. Example: Web application/page. The output should vary depending on the user. If I am a professor teaching Mathematics, I should see something very different from if I am a student taking Mathematics. Provide a “read-me” file detailing how to access and use this interface.