Criterion A—Planning

Defining the problem

My client and advisor, **Mr. Arias Vivas**, is an IB Coordinator at my school. Each year, he collects students' course selections and creates class schedules. Usually, he creates "six groups of classes," and then assign the groups to the available teaching blocks. Because students have distinct course selections (different subjects and levels), it takes "great efforts" for Mr. Arias to generate six groups of classes without conflicts manually. Moreover, he gets imperfect solutions: Some classes contain too few students, and some students do not get their "preferred selections."

Mr. Arias asks me to create a system that helps him with scheduling. After examining the problem and considering my capability, I propose to create a system that helps him **create six groups of classes**.

Rationale for the proposed solution

I have decided to program with Java. The system requires fast automation, since the major problem for my client is that creating schedules are too "time-consuming." I recognize that I need to modulize this complex problem into several sections including inputting data, analyzing data, and outputting data. I choose Java to implement the system because it

1. is license-free and runs on multiple platforms including macOS on which I develop the system and Windows on which my client uses the system;
2. supports graphical interface (GUI)—the system can be more "user-friendly" as required by my client;
3. has multiple I/O functions, so that the system can enable my client to input raw data and view results
4. supports many dynamic data structures that can be stored and manipulated flexibly;
5. is Object-Oriented, so the system can easily modulize the problem by creating interactive classes;
6. supports multithreading, so the system can simultaneously process multiple threads including GUI and scheduling algorithm;

Stating success criteria

After the interviews with my client, we agree that the system should have functions as following:

1. The client can input data from Microsoft Excel files, which contain details of students (names and course selections), teachers (names and subjects) and courses (subjects, levels, and maximum number of classes that can be opened);
2. Generate student groups that are applicable for all the students and teachers in IB (Year 11), specifically:
    1) Generate 6 student groups in total;
    2) Each student has one and only one class in each group;
    3) The size of a class should be more than 3 and less than 20, unless the total number of students choosing the course is less than 3;
    4) If a subject has only one teacher, there cannot be two classes of the subject in the same group;
    5) Separate students of different subjects and HL and SL students of the same subject;
3. Output the results to Microsoft Excel files and text files. Allow the client to see a list of students in each class, a list of classes of each course, and a list of classes in each group.