

## 11. Session no. 11

**Learning goals:** exceptions; usage of recursion (pros and cons) use of simple data structures to store intermediate results to speed-up computations.

1. Improve the program that manages bookings in the Airbus A320 implemented in the previous session. The suggested modifications are:
  - Use exceptions to avoid the program to break when the input data file does not exist or the output file cannot be written.
  - Use exceptions to avoid breaking the program when the format of the data file is not correct. First, try a file with bad format (for example, a line without a comma between the seat and the passenger's name) to see the exception that is launched in this case. Then, use this type of exception to capture and inform the user of this problem without breaking the program.
2. Write a recursive function to compute the  $n$ th value of the Fibonacci series (remember, the series is 1, 1, 2, 3, 5, 8, 13, 21, 34, ...). The first two values of the series are  $f_0 = 1$  y  $f_1 = 1$ , from the third value, the recursion formula is  $f_n = f_{n-1} + f_{n-2}$ .
  - Ask the user for a value of  $n$ .
  - Use a main loop to compute and visualize the result until the user wants to stop (for instance, giving a negative number for  $n$ ).
  - Make experiments to figure out what is the largest input value such that python is able to compute the series using this recursive method in a reasonable amount of time.
3. Analyze the function calls your programs are performing, i.e., try to find an answer to the question: How many times a recursive function is called?

Improve your programs considerably (i.e., the run times) with the help of simple data structures that store intermediate results such that simple look-ups suffice, rather than recalculating values.