**Conference Tack Management**

**Problem Statement:** I selected Conference Track Management Problem for this interview

**Project Structure:**

1. The application is developed in C# .Net 4.5 Framework and Visual Studio 2015
2. CTM is the class library Project which is the algorithm part.
3. CTM.Terminal is the Console Application which is the starting point of the project from where one can pass Input file path and will get the result on the console screen
4. CTM.Tests is the Test Project which tests for CTM project

**Algorithm:**

**Entities:**

***Conference:*** A Conference is set of Events grouped into multiple Tracks or days

***Event***: An Event can be divided into Session and Single Event

***Session***: A Session can have multiple Talks

***Single Event***: A Single Event will have exactly one Talk which will state its actual Start Time and End Time.

***Lunch***: is a Single Event which has Fixed Start Time of 12 PM and End Time of 1 PM

***Networking Event***: A Singe Event can be further divided into Flexible Start Time Event like Networking Event. The Networking Event is Flexible Start Event which has Start Time of 4 PM but can be extended up to 5 PM which depends on its previous Event's last Talk End Time

***Track Format***: Conference takes the Track Format. Track Format is the format of the Conference's Events. In this case the order is

* + Morning Session
  + Lunch
  + Afternoon Session
  + Networking Event

The Track Format executes in the order of addition of Event in its List

It could have any number of Event as long as it is an instance of Event

*Example*: A new Event of Night Session or Tea Event could be added to the list

***Input Processor***: It reads the file, validate Talks and add to Conference. It returns a Conference

***Output Processor***: It prints and write the output

**Workflow**:

* Algorithm checks the type of Event it is processing.
* If it is an Session Event like Morning Session it will process the Talk to Schedule
* The Session will fill its Time Slot in such a way that it has no or less Spare Time.
* Spare Time of an Event is the unused time remaining for that Event.
* So the algorithm tries to keep the Minimum Spare time for the Event
* In this case, Morning Session and Afternoon Session has similar Scheduling approach.
* So it directly uses its parent Schedule method
* But if one want , he could add new Session like Night Session which might implement its own Scheduling approach
* If the Event is Single Event like Lunch , it just add a new Event to list which its default Star Time and End Time
* The Networking Event is also a Single Event but it has variable Start Time of 4 PM to 5 PM
* It Starts time will vary depends upon the last Talk of its previous Session End Time
* If the last Talk has End Time less than 4 PM then it can start from 4 PM
* But if the last Talk End Time is between 4 PM to 5 PM then it can start from when the last Talk ends.
* Conference will go on processing unless all its Talks get Scheduled
* Conference will go on creating multiple Tracks with the same Track Format until all the Talks are scheduled
* Finally, it will have all have Track List which will hold its Events
* The Conference instance could be passed to the Output Processor for printing
* The Output Processor has the facility to save the content in File