**Conference Track Manager – Design Notes**

**Programming Language Used –**

Java

**Assumptions –**

1. The input file contains the talk details in correct order.
2. No parsing issues in the input details – the talks either contain a valid lightning title or a correct time.
3. No empty lines in the input file.
4. The meeting time is always in minutes or lightning and not in hours.

**Design Explanation –**

Constants package contains a class hosting the constants used throughout the program.

Main package contains the main launcher and conference scheduler.

Model package contains the model POJO class to host the talk details – talk title, talk duration and start time.

Exception package contains custom exception – this is used to throw appropriately messaged exception from different flows of the program.

Util package contains utility class to perform different operations –

1. Read and parse the input file and populate talk details.
2. Populate the lunch and networking events – as a one time activity.
3. Print the scheduled conference tracks on the console.

**Instructions to launch the program –**

Provide a complete path to a valid Input file as a program argument.

com.learning.talks.main.Main class contains the “main” method and is responsible for launching the program.

**Program Input –**

The program takes one argument – the complete path of the input file which contains the meeting details.

A reference input file is also provided in the program zip.

**Implementation Logic Details –**

* Given an input set, two tracks are obtained from the scheduler.
  + Talks are scheduled right from the morning in the first Track.
  + Talks are scheduled from evening slots first in the second Track.
* Calendar API is used to track the start times of each talk in the tracks.
* 3 hours i.e. 180 minutes is considered morning slot – as start time is 09:00 a.m. and lunch should start at 12:00 p.m.
* 4 hours i.e. 240 minutes is considered evening slot – as afternoon slot can start from 1p.m. (post lunch) and networking event should start no later than 5 p.m.
* LinkedHashMap is used to maintain the order of scheduled conferences. Talks are stored as key value pairs. A map is used instead of the list in case we want to use another talk attribute (external to the talk details) as the key.
* Morning and Evening slots are maintained and lunch times are appropriately derived based on the progress of talk scheduling.
* The order and duration of talks in the input file will play a role in the slot in which they are scheduled. The attempt is to navigate through each input and fill the appropriate slot as soon as input is processed.
* If lesser talks are available, networking event will still start no earlier than 4:00 p.m.
* If talk durations are lesser (all lightning or lesser than an hour durations), the slots are filled accommodating the maximum number of talks possible.
* Skipped Time entries are entries of talks which are skipped from scheduling.
  + A commented System.out.println() statement captures such talk details. They can be uncommented and enhanced to calculate number and total time of unscheduled talks.

**Possible Enhancements –**

The program can be enhanced to handle the following –

* Invalid input – not containing any time or lightning
* Empty line in input file