WATERLOO
ENGINEERING

Programming Competition
Competition Problem Statement

Spring 2025
Waterloo Engineering Competition
July 6, 2025

Schedule

The schedule of the Fall 2024 Programming competition is as follows:

Time	Event	Location
8:00-8:30	Sign-in	E7 Event Space
8:30-9:00	Welcome to WEC	E7 Event Space
9:00-9:30	Transition to Rooms, Competition Introduction, & Questions Period	E7 4043/4053
9:30-11:00	Competition time	E7 4043/4053
11:00-1:00	Lunch	E7 Event Space
1:00-3:30	Competition time	E7 4043/4053
4:00-6:00	Team presentation and judging	E7 4043/4053
6:00-6:30	Judging Deliberation	E7 Event Space
6:00-7:00	Dinner	E7 Event Space
7:00-7:15	Closing Ceremonies	E7 Event Spacet

*Times are approximate and will be confirmed on the day of the competition.

Note: Lunch will not be a designated break and will overlap with the competition time. Participants are encouraged to manage their time effectively and take breaks as needed while continuing to work on their solutions.

Scenario:

The scenario for this competition is to optimize your time as a student. The biggest challenge most students who come to university first face is how to make the best of their time. From meeting academic goals to fulfilling their social needs to managing their financial and physical security, all of these things take time and are important to meet for a university student. The goal of your completed project is to simulate a student's , social, academic and financial requirements and produce a weekly schedule that meets or maximizes the total possible levels of fulfillment in all these factors.

Objective:

To design and implement a simulation that takes a student's weekly commitments (class, studying, design teams/clubs, social outings, and part time work) and produces the most optimized weekly schedule for the student.

Functional Requirements:

1. The input for the problem is a .txt file named **input.txt**

   a. **Social Activities**

      i. This Section begins with a line of the format: (SOCIAL, S) where S is the number of lines with each line being the constraints for a social activity.

      ii. Possible activities would range from going to the mall, watching a movie, going out for dinner with friends, to doing a social club on campus

      iii. Each line then follows the following **Format: (name, location, onCampus, hours)**.

         1. **Name** will just be a string (i.e. Dinner with friends at BeerTown)
         2. **Location** is also a string (i.e BeerTown)
         3. **OnCampus** will be a true or false value
         4. **Hours** will be in 24-hour time codes (ex. 12:00 – 13:00)

   b. **School**

      i. This Section begins with a line of the format: (SCHOOL, A) where A is the number of lines with each line being the constraints for a school commitment.

      ii. To better your school score you can allocate studying time within your schedule by the hour

      iii. They follow a similar format as the social activities, but with the additional parameter of type which defines whether the school commitment is a class, lecture, lab, or tutorial.

      iv. **Format: (class name, type, location, hours)**

1. If the item is a lecture, lab, or tutorial, the class name value will be the course code (ex. ECE 150). If it's not, it will be a string with the value of "N/A"
2. Type will be the activity: lab, study, class, and so on
3. Location will just be a building code like E7
4. There is no on campus value because all school activities are assumed to be on campus
5. Hours are the same format as in 1.a.iii.4 followed by a two-letter code for the day of the week:
    a. Mo -> Monday
    b. Tu -> Tuesday
    c. We -> Wednesday
    d. Th -> Thursday
    e. Fr -> Friday
    f. Sa -> Saturday
    g. Su -> Sunday

c. Extracurriculars

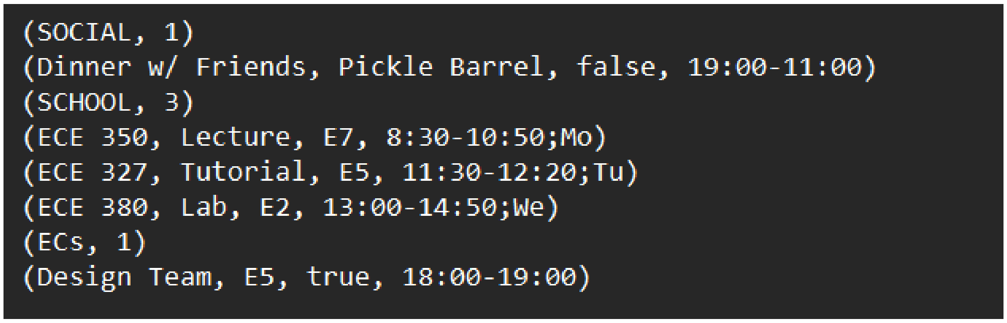
i. This Section begins with a line of the format: (ECs, E) where E is the number of lines with each line is the constraints for an extracurricular.

ii. Format: (type, location, onCampus, hours)

1. Type would be things like your clubs or design teams
2. Location is described as in 1.a.iii.2-4
3. OnCampus defined as in 1.a.iii.3
4. Hours defined as in 1.a.iii.4 without the day code

d. Sample Input:

i. Here's Screenshot of what a sample input file would look like:



```
(SOCIAL, 1)
(Dinner w/ Friends, Pickle Barrel, false, 19:00-11:00)
(SCHOOL, 3)
(ECE 350, Lecture, E7, 8:30-10:50;Mo)
(ECE 327, Tutorial, E5, 11:30-12:20;Tu)
(ECE 380, Lab, E2, 13:00-14:50;We)
(ECs, 1)
(Design Team, E5, true, 18:00-19:00)
```

2. Some other information
    a. Travel time is considered when going from location to location, make sure to keep travel time blocked in your output

i. When traveling from location to location within the campus, we're taking an average of 15 minutes of travel time (i.e E7 -> DWE is taken as 15 minutes)

ii. When traveling to and from locations outside of campus, we're assuming 25 minutes of travel time (i.e E7 -> Conestoga, Trapped -> Pickle Barrel, Laurier campus -> E5)

iii. The onCampus parameter is there to clarify which travel time needs to be considered

iv. There is no travel time associated with studying unless when going to a social commitment (i.e Studying -> Escape Room) and the travel time follows the same rules as described in **2.a.ii**

b. Every commitment is optional, meaning your code can skip them if you believe that produces an overall better output (**ONLY IN THIS SCENARIO, class is very important, don't skip)**

c. Inputs without a day code can be placed in the schedule wherever you wish

d. Mandatory sleep is required from 11PM to 7 AM every day, and being an hour or more off of this time span has a score reduction of 5 (i.e. if the sleep span is 10:10PM to 7:10 AM then there is –5 on the final composite score)

e. **SCORING:** Each type of commitment starts off with a base score of 10 (and EC metric, a SCHOOL metric, and a SOCIAL metric)

i. School Scoring System:

1. Skipping out on Lecture -> -2
2. Skipping Lab -> -1.5
3. Skipping tutorial -> -0.75
4. Completing any of the above listed is an increase in the value by the same amount
5. Studying has its own set of rules:

a. +0.5/hour studied

b. - 3 for not studying at least 5 hours in the week

ii. Except for your SCHOOL score, completing an activity under the other categories increases its metric by 1

iii. At the end, a composite score is made by the following formula:

1. SCHOOL + 0.75 * EC + 0.5 * SOCIAL

iv. You are responsible for keeping track of each score throughout the execution of your program and displaying the final calculated composite score

Expected Output:

- Your code is expected to output the week's schedule to a json file with the following format:

```
{
Monday: {
Activity1_Name: [Start time, end time],
Activity2_Name: [Start time, end time],
                  .......
        },
Tuesday: {
Activity1_Name: [Start time, end time],
Study: [Start time, end time],
        },
Wednesday: {
Activity1_Name: [Start time, end time],
Study: [Start time, end time],
Activity2_Name: [Start time, end time],
        },
Thursday: {
Activity1_Name: [Start time, end time],
Activity2_Name: [Start time, end time],
                  .......
        },
Friday: {
Activity1_Name: [Start time, end time],
Activity2_Name: [Start time, end time],
                  .......
        },
Saturday: {
Activity1_Name: [Start time, end time],
                  .......
        },
Sunday: {
Activity1_Name: [Start time, end time],
Study: [Start time, end time],
        },
Score: 30
}
```

Deliverables

You will have 6 hours to design and build your solution. At the 6-hour mark, your team must provide all outputs from the programs, your finalized code and any presentation material.

All your code needs to be in your GitHub repository's Main branch before the 6-hour deadline for it to be considered. Basic instructions on how to execute your code must be provided (Eg. An updated readme). You must specify the language and version your code uses. You can assume that anyone who runs the code has the appropriate languages installed. A list of required packages is advised. (e.g. Pandas, Gurobi).

Presentation Deliverable

Only presentations submitted in your GitHub repository Main Branches before the 6-hour deadline will be used in the presentation period. No work may be done on the presentation deliverable after the deadline has passed

You may discuss your team's implementation for each algorithm.

Presentation Rules

Competitors are allotted a maximum of 10 minutes to present their solutions, during which all team members must be present and actively participate. Failure to do so will result in a penalty from the judges. Following the presentation, judges will have up to 5 minutes to ask questions.

Presentation order will be determined randomly and announced two hours before the presentations begin. All teams must be present for this announcement, and switching presentation slots is only permitted in cases of emergency, or at the discretion of the competition director. A visible countdown timer will be provided. Presentations will be conducted privately, without an audience, and teams will receive feedback forms after the winners are announced.

Judging Matrix

A panel of judges will evaluate the teams based on the following criteria:

Execution	__ / 65	
Composite Score	25	
Number of Activities/Tasks Addressed	15	
Clean and Accurate Schedule Produced	15	
Solution Produces an Accurate Score	5	
Solution Is able to Read Input Without Errors	5	
Code Design and Documentation	__ / 10	
Structure	5	
Comments	5	
Presentation	__ / 25	
Design process and justification	10	
Presentation quality and Speaker Variability	5	
Responses to questions	5	
Visual Aids	5	
Deductions		
Presentation Goes Over Time	-2 per minute	
Total	__ / 100	