# Homework 2
CENG431 – Building Software Systems

In this homework you are expected to implement and design a "***Uncontrolled Junction Simulation***" in Java. You should fulfill the concepts of:

- Mediator Design Pattern
- UML Class Diagrams
- UML Sequence Diagrams
- UML State Diagrams

1. In this application you are expected to implement a simulation of a 4-way uncontrolled intersection. In the simulation 20 vehicles will be randomly generated and the simulation will show the user right-of-way for the vehicles.

2. Vehicles for the simulation will be as shown in the table:

Vehicle	Motor Type	Generation Probability
Car	Motor Vehicle	35%
Bus	Motor Vehicle	10%
Truck	Motor Vehicle	10%
Motorcycle	Motor Vehicle	15%
Tractor	Motor Vehicle	10%
Engineering Vehicle	Motor Vehicle	5%
Bicycle	Non-motorized Vehicle	10%
Horse-drawn Vehicle	Non-motorized Vehicle	5%

3. Roads for the intersection should be numbered from 1 to 4 and should be in clockwise:

```
              1
      4               2
              3
```
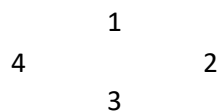
4. Steps for generating vehicles should be as follows:
    a. First generate a vehicle according to generation probability which is given in the table above.
    b. Then decide randomly on which road the generated vehicle will be placed. There are four roads as given above and each have the same probability for placement.
    c. Lastly decide randomly the direction of the vehicle. The vehicle may want to go straight, turn right, or turn left.

5. The rules for the right-of-way are as follows:
    a. Non-motorized vehicle should give way to motor vehicles no matter their directions.
    b. Tractors or engineering vehicles should give way to other motor vehicles no matter their directions.

c. Vehicles should give way to traffic approaching from the right (For example: a car on the 3rd road should give way to a car on the 2nd road).

d. Vehicles should give way if they will turn left (For example: a car on the 2nd road should give way to a car on the 3rd road if the car on the 2nd road will turn left. However, if both cars will turn left, the rule in the section c is applied).

e. All vehicles should give way to pedestrians.

f. If there is a case happen where all the vehicles have the same passage right, then give the right-of-way to the vehicle on the 1st road.

g. If there is a case happen where both vehicles on the roads have the same passage right, then give the right-of-way to the vehicle on the road with smaller index number (e.g., in case with 2nd and 4th roads, then give the right-of-way to the vehicle on the 2nd road)

6. Simulation will work round by round. In each round a vehicle will go to its chosen direction. In each round there is possibility of generation of a pedestrian which is 10%. If a pedestrian is generated, their road of crossing also will be decided randomly (e.g., 4th road). A pedestrian will cross the road in 2 rounds.

7. You are expected to print the short results to the console after simulation is completed. Short result will be as in the example output:

Vehicles:
    1st road:
            1. Car (Go Straight)
            2. Car (Turn Right)
            3. Bus (Turn Left)
            4. Bicycle (Go Straight)
            5. Truck (Turn Left)
            6. Car (Turn Left)
    2nd road:
            1. Car (Turn Left)
            2. Motorcycle (Turn Right)
            3. Engineering vehicle (Go Straight)
            4. Car (Turn Left)
            5. Bicycle (Turn Left)
    3rd road:
            1. Bus (Go Straight)
            2. Car (Turn Left)
            3. Tractor (Turn Right)
            4. Car (Turn Right)
    4th road:
            1. Car (Turn Left)
            2. Car (Go Straight)
            3. Horse-drawn vehicle (Turn Right)
            4. Motorcycle (Turn Left)
            5. Car (Go Straight)

    Right-of-way:
    1. Car on the 1st road went straight
    2. Car on the 1st road turned right

3. Bus on the 3rd road went straight
4. Bus on the 1st road turned left
5. Car on the 2nd road turned left

A pedestrian is waiting on the 2nd road

6. Car on the 3rd road turned left

The pedestrian is crossing on the 2nd road

7. Car on the 4th road turned left
8. Motorcycle on the 2nd road turned right

.

.

.

8. You are also expected to save extended results to a text file SimResult.txt as follows:
   a. Generation information for each vehicle
   b. Vehicles on the junction will be printed for each round (as in the short results, but for each round a vehicle will not be on the list since it left the junction in the previous round).

9. You are expected to draw a **UML Class diagram**, a **UML Sequence diagram**, and a **UML State diagram** for your program.

Important Notes:

1. **Do NOT request inputs in your app**. Printing the results of the queries will be enough.

2. You should use relative paths (e.g., Files/sample.csv) instead of absolute paths (e.g., C:\\user\\eclipse-workspace\\MyProject\\Files\\sample.csv).

3. To support **Turkish characters,** you may need to change your project's text file encoding to UTF8: Right click on your project (in package explorer) → Properties → Text file encoding → Other → UTF8 → Apply.

4. You are expected to write clean, readable, and tester-friendly code. Please try to maximize reusability and prevent from redundancy in your methods.

Assignment Rules:

1. In this lecture's homework, there are no cheating allowed. If any cheating has been detected, they will be graded as 0 and there will be no further discussion on this.

2. You are expected to submit your homework in groups. Therefore, only one of you will be sufficient to submit your homework.

3. Make sure you export your homework as an **Eclipse project**. You can use other IDEs as well, however, you must test if it supported by Eclipse. If the project import is not accomplished in Eclipse, you will lose points.

4. Submit your homework through Cloud-LMS.

5. Your exported Java Project should have the following naming format with your assigned group ID (which will be announced on MS Teams) as the given below:

Also, the zip folder that your project in should have the same name

6. Please beware that if you do not follow the assignment rules for exporting and naming conventions, you will lose points.

7. Please be informed that your submissions may be anonymously used in software testing and maintenance research studies. Your names and student IDs will be replaced with non-identifying strings. If you do not want your submissions to be used in research studies, please inform the instructor (Dr. Tuglular) via e-mail.