# ARTIFICIAL INTELLIGENCE PROJECT

# TIME TABLE GENERATOR

Basit Ali (21k4510)
Muhammad Jahanzaib (21k3361)
Raja Inam Abbasi (19k0234)
Wajheeh Khan (21k3227)

**Introduction:**
This project report describes the implementation of a timetable generator using a genetic algorithm. The genetic algorithm is a heuristic optimization technique that mimics the process of natural selection in order to find the optimal solution to a problem. The project uses Python programming language to develop the algorithm, and pretty table library for displaying the generated timetable in a tabular format.

**Problem Analysis:**
The problem statement for this project is to generate a timetable that satisfies a set of constraints. The constraints are related to the availability of resources such as rooms, instructors, and meeting times, as well as the scheduling of courses offered by different departments of a university. The solution to this problem is a timetable that assigns courses to different meeting times, rooms, and instructors, and satisfies all the constraints.

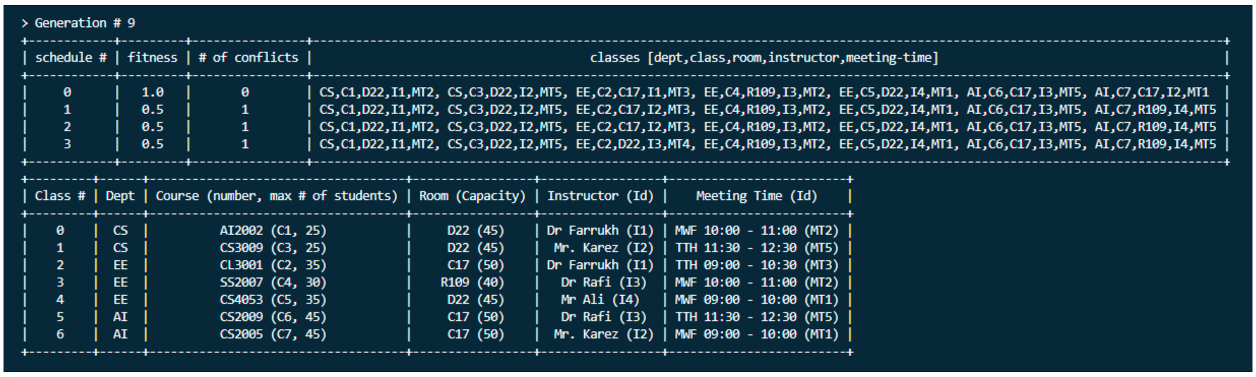
**Solution Design:**
The approach taken to solve the problem is to use a genetic algorithm. A genetic algorithm is an optimization technique that simulates the process of natural selection. It starts with a population of random solutions, and iteratively applies genetic operators such as selection, crossover, and mutation to generate new solutions. The fitness of each solution is evaluated based on how well it satisfies the constraints. The algorithm terminates when a solution that satisfies all the constraints is found or a maximum number of generations is reached.

**Implementation & Testing:**
The implementation of the genetic algorithm is done using Python programming language. The implementation is divided into several classes that represent different entities and functionalities of the algorithm. The classes used in the implementation are Data, Room, Instructor, Meeting Time, Course, Department, Class, and Schedule.
The Data class contains information about the resources available for scheduling. It defines the rooms, instructors, meeting times, courses, and departments. The Room, Instructor, Meeting Time, Course, and Department classes define the attributes and functionalities of these entities. The Class class defines a class,

which is a combination of a course, instructor, room, and meeting time. The Schedule class defines a schedule, which is a collection of classes. The Schedule class also defines the genetic operators used in the genetic algorithm, such as selection, crossover, and mutation.

The genetic algorithm starts with a population of random schedules. The fitness of each schedule is evaluated based on how well it satisfies the constraints. The fitness function counts the number of conflicts in each schedule, where a conflict occurs when a class is assigned to a room with insufficient seating capacity. The selection operator selects the best schedules from the population based on their fitness. The crossover operator generates new schedules by combining the best schedules from the population. The mutation operator randomly changes the attributes of the classes in a schedule. The algorithm iteratively applies these genetic operators to generate new schedules, until a solution that satisfies all the constraints is found or a maximum number of generations is reached.

**Results:**
The genetic algorithm was implemented and tested using the given code. The algorithm was able to generate a timetable that satisfies all the constraints for the given problem. The generated timetable is displayed in a tabular format using the pretty table library. The timetable shows the classes assigned to different meeting times, rooms, and instructors.



```
> Generation # 9
+------------+---------+-----------------+------------------------------------------------------------------------------------------------------------------------------+
| schedule # | fitness | # of conflicts |                                         classes [dept,class,room,instructor,meeting-time]                                     |
+------------+---------+-----------------+------------------------------------------------------------------------------------------------------------------------------+
|     0      |   1.0   |       0         | CS,C1,D22,I1,MT2, CS,C3,D22,I2,MT5, EE,C2,C17,I1,MT3, EE,C4,R109,I3,MT2, EE,C5,D22,I4,MT1, AI,C6,C17,I3,MT5, AI,C7,C17,I2,MT1  |
|     1      |   0.5   |       1         | CS,C1,D22,I1,MT2, CS,C3,D22,I2,MT5, EE,C2,C17,I2,MT3, EE,C4,R109,I3,MT2, EE,C5,D22,I4,MT1, AI,C6,C17,I3,MT5, AI,C7,R109,I4,MT5 |
|     2      |   0.5   |       1         | CS,C1,D22,I1,MT2, CS,C3,D22,I2,MT5, EE,C2,C17,I2,MT3, EE,C4,R109,I3,MT2, EE,C5,D22,I4,MT1, AI,C6,C17,I3,MT5, AI,C7,R109,I4,MT5 |
|     3      |   0.5   |       1         | CS,C1,D22,I1,MT2, CS,C3,D22,I2,MT5, EE,C2,D22,I3,MT4, EE,C4,R109,I3,MT2, EE,C5,D22,I4,MT1, AI,C6,C17,I3,MT5, AI,C7,R109,I4,MT5 |
+------------+---------+-----------------+------------------------------------------------------------------------------------------------------------------------------+
+---------+------+-----------------------------------+-----------------+-----------------+-------------------------+
| Class # | Dept | Course (number, max # of students) | Room (Capacity) | Instructor (Id) |    Meeting Time (Id)    |
+---------+------+-----------------------------------+-----------------+-----------------+-------------------------+
|    0    |  CS  |          AI2002 (C1, 25)          |    D22 (45)     | Dr Farrukh (I1) | MWF 10:00 - 11:00 (MT2) |
|    1    |  CS  |          CS3009 (C3, 25)          |    D22 (45)     |  Mr. Karez (I2) | TTH 11:30 - 12:30 (MT5) |
|    2    |  EE  |          CL3001 (C2, 35)          |    C17 (50)     | Dr Farrukh (I1) | TTH 09:00 - 10:30 (MT3) |
|    3    |  EE  |          SS2007 (C4, 30)          |   R109 (40)     |   Dr Rafi (I3)  | MWF 10:00 - 11:00 (MT2) |
|    4    |  EE  |          CS4053 (C5, 35)          |    D22 (45)     |   Mr Ali (I4)   | MWF 09:00 - 10:00 (MT1) |
|    5    |  AI  |          CS2009 (C6, 45)          |    C17 (50)     |   Dr Rafi (I3)  | TTH 11:30 - 12:30 (MT5) |
|    6    |  AI  |          CS2005 (C7, 45)          |    C17 (50)     |  Mr. Karez (I2) | MWF 09:00 - 10:00 (MT1) |
+---------+------+-----------------------------------+-----------------+-----------------+-------------------------+
```

**Conclusion:**
In conclusion, the timetable generator using a genetic algorithm is an effective approach for generating timetables that satisfy a set of constraints. The genetic algorithm uses the principles of natural selection to iteratively generate new schedules until a solution that satisfies all the constraints is found. The implementation of the genetic algorithm using Python 3 programming language and pretty table library was successful in generating a timetable that satisfies all the constraints for the given problem. The generated timetable can be used as a basis for scheduling courses offered by different departments of a university.