# Student Enrollment Simulation Documentation

## Overview

The student enrollment simulation is a discrete event simulation that models the process of students enrolling in courses at a fictional educational institution. The simulation uses a priority queue to manage events, such as student arrivals and course capacity changes, in chronological order. It tracks various statistics like server idle time, average waiting time, and the probability that a student will wait in line.

## Classes

### Event

A class representing an event in the simulation.

#### Attributes

* event\_type (str): The type of event, either "Student Arrival" or "Capacity Change."
* event\_time (float): The time at which the event occurs.
* event\_parameters (tuple): Additional parameters related to the event, such as student information or course capacity details.

### Student

A class representing a student.

#### Attributes

* student\_id (str): Unique identifier for the student.
* name (str): Name of the student.
* age (int): Age of the student.
* enrolled\_courses (list): List of courses in which the student is enrolled.

### Course

A class representing a course.

#### Attributes

* course\_code (str): Unique code for the course.
* course\_name (str): Name of the course.
* max\_capacity (int): Maximum capacity of the course.
* current\_enrollment (int): Current number of students enrolled in the course.
* enrolled\_students (list): List of students enrolled in the course.

## Simulation Workflow

1. Initialize the simulation with a list of courses, enrolled students, and an empty event queue.
2. Generate initial events for student arrivals with random arrival times and course choices.
3. Simulate events one by one from the event queue, handling student arrivals and course capacity changes.
4. Calculate the server idle time and total waiting time during the simulation.
5. Track statistics, such as the average waiting time and the probability that a student will wait in line.
6. Output the enrollment status for each student and display the simulation results, including the average waiting time, probability to wait in line, and server idle time.

## Functions

### initialize\_simulation()

Initialize the simulation by creating the list of courses, enrolled students, and the event queue. Generate initial events for student arrivals.

### simulate\_events(courses, enrolled\_students, event\_queue)

Simulate events by processing student arrivals and course capacity changes. Calculate the server idle time and total waiting time.

#### Parameters

* courses (list): List of Course objects representing available courses.
* enrolled\_students (list): List of Student objects representing enrolled students.
* event\_queue (list): Priority queue of Event objects containing events to be processed.

### track\_simulation\_statistics(enrolled\_students, server\_idle\_time, total\_waiting\_time, students\_waited)

Calculate statistics, such as the average waiting time and the probability that a student will wait in line.

#### Parameters

* enrolled\_students (list): List of Student objects representing enrolled students.
* server\_idle\_time (float): Total time the server was idle during the simulation.
* total\_waiting\_time (float): Total waiting time for students who had to wait in line.
* students\_waited (int): Number of students who had to wait in line.

### output\_results(enrolled\_students, average\_waiting\_time, probability\_to\_wait, server\_idle\_time)

Display the enrollment status for each student and output the simulation results, including the average waiting time, probability to wait in line, and server idle time.

#### Parameters

* enrolled\_students (list): List of Student objects representing enrolled students.
* average\_waiting\_time (float): Average waiting time for students who had to wait in line.
* probability\_to\_wait (float): Probability that a student will wait in line.
* server\_idle\_time (float): Total time the server was idle during the simulation.

### simulate\_student\_enrollment()

The main function to run the student enrollment simulation. Calls the other functions in the correct sequence.