## Project No. 1

## Sleeping teaching assistant

PROJECT DESCRIPTION

The project is a simulation of a scenario where multiple students seek help from a teaching assistant (TA). The TA can only help one student at a time, and there are a limited number of chairs outside the TA's office where students can wait. The project uses multithreading and semaphores to manage the interactions between students and the TA.

WHAT HAS BEEN DONE

The project is implemented in Java and uses the java.util.concurrent.Semaphore class for synchronization. The main classes in the project are Student, TeachingAssistant.

1. `TeachingAssistant`: The TeachingAssistant class represents a TA who alternates between helping students and sleeping. The TA waits for a signal from a student to wake up. When they receive a signal, they work with the student for a certain amount of time. If there are other students waiting in chairs, they then work with those students , This class implements the Runnable interface and represents a teaching assistant
2. `Student`: The Student class represents a student who alternates between programming and seeking help from the TA. When a student needs help, they check if the TA is available. If the TA is available, the student wakes up the TA and works with them for a certain amount of time. If the TA is not available, the student checks if there are any chairs available. If a chair is available, the student waits for the TA to become available. If no chairs are available, the student goes back to programming.
3. `\_work`: This class provides a graphical user interface for the simulation. It allows the user to specify the number of TAs, students, and chairs, and then run the simulation.

## THE MAIN COMPONENTS OF THE PROJECT ARE:

.

1. `TeachingAssistant` Class: This class represents a teaching assistant (TA). It implements the `Runnable` interface and has a `run` method that defines the behavior of the TA when they are awake and when they are sleeping.
2. `Student` Class: This class represents a student. It also implements the `Runnable` interface. The `run` method defines the behavior of the student when they are programming and when they are asking for help.
3. `work` Class: This class is responsible for the graphical user interface of the application. It allows the user to input the number of TAs, students, and chairs, and then run the simulation.
4. Semaphores: These are used to manage the chairs outside the office and to determine if the TA is available. They are key to synchronizing the interaction between the TA and the students .

Member Rule:

زياد محمد حمدان , عمر هاشم محمد : : Gui ,Created the Student class

عبدالرحمن عادل محمد : Mutexlock

احمد رجب احمد : documentation

اسراء محمد محمود : Created the TA class