

## **CECS 326 – Project #2 Report**

### ***The Design Explanation:***

- **DiningPhilosophers.java:**
  - For this file, we create a program within the file that lays out the dining philosopher problem. Within our class, DiningPhilosophers, we create 3 main functions that create the dining server problem, then another one that creates the philosopher threads, and once compiled, the program starts by working on philosopher threads
- **DiningServerImpl.java:**
  - For this file, we import the Lock, Condition & ReentrantLock classes from the locks class. Within the DiningServerImpl class, we create multiple private and public classes for specific tasks. This includes creating arrays for locks and conditions, initializing locks and conditions for each fork, and then taking & putting away a fork depending on the user.
- **DiningServer.java:**
  - For this file, we create a class that contains the philosopher's methods when being called upon during compiling. We create two functions that are called during specific times such as when we want to eat and when the user is finished eating.
- **Philosopher.java:**
  - For this file, we created the philosopher class to help represent each philosopher thread and distinguish between the one that is thinking and eating. We create many random and public variables to identify their purpose in each function we create when it comes to the philosopher running, thinking, and eating.

***Project Demonstration YouTube Link:***

- <https://youtu.be/4BWgOzW8eAY>
- This video introduces who we are and demonstrates how our completed program works.

***Project Contributions:***

- When it came to contributing to the project, we both evenly contributed to the assignment including the creating/debugging of the files.
- In regards to the uploading and creating of the YouTube demonstration video, Alina was in charge of orchestrating and compiling the project during the recording session.
- In addition, Miles created the README.md file that helps to instruct the user on how to run the project that we have completed.