**Synchronization steps:**

1. Set up customer containers and the containers that required in Simulation.java.
2. Initiate the cook threads before all customers initiated.
3. After being initiated, all customers start to submit the order.
4. Cook identify the food’s type then submit the food to specific machines one by one(using synchronized).
5. Machines receive food list one by one and make foods one by one(using synchronized), then tell cook that the food has been cooked.
6. Cook record the food that has been cooked then finish the order according to the priority.
7. Cook would add the customer to the completed list after they finish the order.
8. Customers threads wait until their orders being settled(notify all).
9. Once customer’s order being completed, customer would leave coffee shop, then new customer would come in.

**Cook responsibilities:**

1. Receive orders from different customers, then Iterate the order according to the priorities.
2. Identify and submit the food in the order to machines.
3. Record finished orders.

**Customer responsibilities:**

1. Submit orders to cook.
2. Check if the order has been placed or not. If current customer’s order has been placed, the customer would leave from the coffee shop.

**Machine responsibilities:**

1. Receive food from cook then cook the food.
2. Tell cook wait if the machine is filled.
3. Notify cook food finished by add the current food to the “foodsDone” list.

Simulation:

1. Manage the whole process and initiate required data structure.