**Write-Up**

**Step 5**: I need between 100 to 105 waiting chairs to have all 200 customers served without any drops. The greater the number of waiting chairs, the better the chances of getting 0 customer drops.

**Step 6**: I need 5 barbers to serve all 200 customers without any waiting or have customers being dropped. I got this number by trying all combinations until 5, which consistently gave 0 customers dropped.

**Illustration**

A white paper with writing on it

Description automatically generated

**Analysis**

Mutex locks are used to make sure only one thread at a time is executed.

**From the customer’s perspective**: When a customer arrives at the shop, they attempt to find a barber or a waiting chair. If no chairs are available and all barbers are busy, they leave (custDrops increments). If there are no barbers immediately available, the customer takes a waiting chair and waits for a barber to wake them. They notify their barber when they are ready for service. Once a barber is free, the customer moves to the barber’s service chair and waits for the haircut to finish (gettingHaircut flag). They leave after paying the barber (paid flag).

**From the barber’s perspective**: A barber starts by checking if they have a customer. If not, they sleep (added to available\_barbers\_ queue). When woken by a customer, the barber starts the haircut. They wait for the customer to indicate readiness (gettingHaircut) before proceeding. Once the haircut is done, the barber signals the customer for payment and ensures payment completion before handling the next customer. If no customers are waiting, the barber returns to sleep by going back into the queue.

Each customer has a condition variable to synchronize with the barber assigned to them. This ensures they wait for a barber to become available and receive signals at appropriate stages (service start and payment). Each barber has a condition variable to wake them up when assigned a customer or when waiting for the customer's payment.