**Examples of Starvation:**

For this scenario problem of starvation will occur if the customers don’t follow any order for getting a haircut, as some won’t get a haircut even though even after waiting for a long time.

* -When the barber finishes a haircut, he inspects the waiting room to see if there are any waiting customers and falls asleep if there are none

**How did solve the starvation:**

To handle this problem in my code I have inserted the customers in a linked list which follows the first in first out property. So, every time a customer sits in a waiting room, they will be selected by the barber in first come first serve basis. We could have also used other data structures like a stack, but the linked list seems like the best choice for this scenario.

**Explanation for real world application and how did apply the problem:**

Initially when there is no customer on-call all call-executives just relax and wait for the call. The moment the first customer dials the number he/she is connected to any call-executive and in a scenario when all call-executives are busy the customer will have to wait in a queue till they are assigned to a call-executive. If all executives are busy and the waiting line is full, the customers are disconnected with a message that executives are busy and customers will be contacted later by the company. This best relates to this design as the customers are picked from the queue in a first come first serve basis and call-executives are utilized in such a way that everyone executive gets at least one call.

In this scenario we can have the following design similarities:

1. The critical section will be the call between executive and customer

2. The waiting room will be the waiting queue over a call, where customers will be held in a FIFO manner.

3. Locks can be acquired on the waiting queue so that no two executives pick the same customer.