```
// adjust price based on number of tickets left
def computePrice(ticketsRemaining: Int): Float


// called from purchaseTicket & displayEvent
def getTicketCount(event: UUID): Int =
  // use weak consistency for performance
  readWeak(event+"ticket_count")


def purchaseTicket(event: UUID) = {
  val ticket = reserveTicket(event)
  val remaining = getTicketCount(event)
  // compute price based on inconsistent read
  val price = computePrice(remaining)
  display("Enter payment info. Price: ", price)
}
```