## Concurrent & Parallel Programming

**Question 1**

In the Prado museum in Madrid there is a small seating area where visitors can view Picasso’s Guernica. Each visitor must find a seat and may view the painting for as long as they choose. Your task is to write a class called ViewingStand that manages access to the seats. The class has two methods: findSeat and leaveSeat. Seats may not be shared between visitors, i.e. two visitors cannot share the same seat. In the event that a seat is not available a visitor simply continues to check for one. There is no wait queue. It is important to optimize the search for a seat so that multiple visitors can concurrently search for seats.

Write a simple simulator for your class that uses threads to model visitors.

**Question 2**

Given below is the class SharedArray that uses *fine-grained* locking to optimse parallel access for threads assigning new values to the array. We want to write a method sum that calculates the sum of the values in the array. To do this we cannot allow multiple threads to assign values at the same time. Your task is to write the method sum that is thread safe and does not compromise the *fine-grained* solution used in the assign method. (Hint: Think!)

class SharedArray{

private int data[];

private Lock locks[];

public SharedArray(int f[]){

data = new int[f.length];

locks = new ReentrantLock[f.length];

for(int j = 0; j < f.length;j++){

data[j] = f[j];

locks[j] = new ReentrantLock();

}

}

public void assign(int j, int x){ //assume 0 <= j < data.length

locks[j].lock();

try{

data[j] = x;

}finally{locks[j].unlock();}

}

public int sum(){ }

}