

GRIFFITH COLLEGE DUBLIN

**QUALITY AND QUALIFICATIONS IRELAND
EXAMINATION**

POSTGRADUATE DIPLOMA IN SCIENCE IN COMPUTING

PARALLEL AND DISTRIBUTED PROGRAMMING

Module Code: PGDC-PDP

MASTER OF SCIENCE IN COMPUTING

PARALLEL AND DISTRIBUTED PROGRAMMING

Module code: MSCC-PDP

**POSTGRADUATE DIPLOMA IN SCIENCE IN BIG DATA MANAGEMENT
AND ANALYTICS**

PARALLEL AND DISTRIBUTED PROGRAMMING

Module Code: PGDBD-PDP

MASTER OF SCIENCE IN BIG DATA MANAGEMENT AND ANALYTICS

PARALLEL AND DISTRIBUTED PROGRAMMING

Module code: MSCBD-PDP

Lecturer(s):

Osama Abushama

External Examiner(s):

Dr Mubashir Husain Rehmani

Dr William Clifford

Date: 13th August 2024

Time: 2.15-5.15

**THIS PAPER CONSISTS OF FIVE QUESTIONS
FOUR QUESTIONS TO BE ATTEMPTED
ALL QUESTIONS CARRY EQUAL MARKS**

QUESTION 1

- (a) Explain the difference between a process and a thread. In the case of threads, why is it simple to share data between different threads? **(5 marks)**
- (b) Explain the differences between semaphores and monitors. **(5 marks)**
- (c) Explain the difference between MPI_Send, MPI_Ssend and MPI_Bsend in MPI? **(15 marks)**

Total (25 marks)

QUESTION 2

- (a) Give a high level description of the algorithm for the solution of the Producer-Consumer problem for finite buffers using semaphores. **(5 marks)**
- (b) Using semaphores, write code in Java to implement a solution to the Producer Consumer Problem. You must create a generic class called buffer which has methods get () and put () to retrieve and append integers onto a buffer. (The Buffer need only store one integer at a time). **(20 marks)**

Total (25 marks)

QUESTION 3

- (a) What is the output of the following program

(5 marks)

```
public class MultiThreading {  
  
    private static class MyThread extends Thread {  
  
        public MyThread(String name) {  
            super(name);  
        }  
  
        @Override  
        public void run() {  
            System.out.println(Thread.currentThread().getName());  
        }  
    }  
  
    public static void main(String[] args) {  
        MyThread myThread = new MyThread("myThread");  
        myThread.run();  
    }  
}
```

- (b) In Java, define and explain the three functions of synchronization?

(10 marks)

- (c) What are the 4 necessary and sufficient conditions required for a deadlock to occur in a multithreaded program? **(10 marks)**
Total (25 marks)

QUESTION 4

An automated booking system for a theatre is required. The system must support multiple users that are allowed to book a single seat at a time from a list of available seats. The system must ensure that double booking is not permitted whilst allowing clients free choice of available seats. This means that a seat may appear to be free although it may be booked or in the process of being booked by another client. Implement a class that allows users to book seats whilst not permitting double booking. Two methods needed for this solution.

```
public int[] getSeats(){ ... }  
public boolean bookSeat(int k){ ... }
```

- (a) Implement a **fine-grained** solution for bookSeat method. **(10 marks)**
(b) Can the method bookseat deadlock? Explain. **(5 marks)**
(c) Explain what is meant by false sharing and suggest a solution. **(10 marks)**
Total (25 marks)

QUESTION 5

- (a) Optimize the following code by removing redundant barriers:

```
#pragma omp parallel private(i)  
{  
    #pragma omp for  
    for(i=0;i<n;i++)  
        a[i] += b[i];  
    #pragma omp for  
    for(i=0;i<n;i++)  
        c[i] += d[i];  
    #pragma omp barrier  
    #pragma omp for reduction(+:sum)  
    for(i=0;i<n;i++)  
        sum += a[i] + c[i];  
}
```

- (b) Write a C OpenMP code to sum a large array over N processors. **(10 marks)**
(c) Write a C MPI code to sum a large array over N processors. **(10 marks)**
Total (25 marks)