

GRIFFITH COLLEGE DUBLIN

**QUALITY AND QUALIFICATIONS IRELAND
EXAMINATION**

MASTER OF SCIENCE IN BIG DATA MANAGEMENT AND ANALYTICS

PARALLEL AND DISTRIBUTED PROGRAMMING

Module code: MSCBD-PDP

ONLINE EXAMINATION

Lecturer(s):

Osama Abushama

External Examiner(s):

Dr Joseph Rafferty

Date: 9th February 2021

Time: 2.15-5.15

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

HONOUR CODE

By submitting my exam script via Moodle, I certify that my answers contained in this Examination Script document are entirely composed of my own original work.

During the exam period, which began when I received the exam paper document, I did not work with anyone else on the exam or discuss the examination with anyone else.

I did not copy and hold out as my own any material belonging to or produced by another person.

I understand that failure to adhere to these instructions shall be an Honour Code violation.
Violation of the Honour Code will result in being charged with academic misconduct.

QUESTION 1

- (a) What do we understand by the term race condition? **(5 marks)**
- (b) What is the difference between notify() and notifyAll()? **(5 marks)**
- (c) Implement a Thread-pool concurrent program that uses a callable object to compute frequency of negative even number inside a large array of size 1,000,000 random integers ranging between -100 to 100. Your program should distribute workload between all threads and use optimal number of threads to provide best performance. **(15 marks)**
- Total (25 marks)**

QUESTION 2

The programming language Java provides for monitors by means of locks and condition variables. Locks can be declared and initialized by means of

```
java.util.concurrent.locks.Lock lock = new java.util.concurrent.locks.ReentrantLock();
```

Condition variables are obtained from a given lock by means of

```
java.util.concurrent.locks.Condition c1 = lock.newCondition();
```

- (a) Explain how locks are used to implement mutual exclusion synchronization. **(5 marks)**
- (b) Explain what happens when a program executes c1.await(). **(5 marks)**
- (c) Explain what happens when a program executes c1.signal(). **(5 marks)**
- (d) Explain what is meant with signal and continue in this context. **(5 marks)**
- (e) Explain how normal monitors/object can be used without condition variables? **(5 marks)**
- Total (25 marks)**

QUESTION 3

- (a) A parking house has place for 100 cars. There are many turnstiles (mechanical gates) letting cars in and out. As cars come in a shared counter is incremented. As cars leave the house, the shared counter is decremented. Implement the shared counter as a monitor in Java. See to it that if there are 107 cars in the house, processes trying to increment the counter get delayed until some car leaves the house

(15 marks)

- (b) Explain why the following program prints different totals for the given array. Rewrite it so that this does not happen.

```
class ShowInterf {  
    public static void main(String args[]){  
        int[] b = {23, 45, 33, 13, 28, 45, 34, 54, 67};  
        (new Thread(new Summer(b))).start();  
        (new Thread(new Swapper(b))).start();  
    }  
}  
class Swapper implements Runnable {  
    int b[];  
    public Swapper(int k[]){ b = k; }  
    public void run() {  
        int j, k, t;  
        while (true) {  
            j = (int)(Math.random()*b.length);  
            k = (int)(Math.random()*b.length);  
            t = b[j]; b[j] = b[k]; b[k] = t;  
        }  
    }  
}  
class Summer implements Runnable {  
    int b[];  
    public Summer(int k[]){ b = k; }  
    public void run() {  
        while (true) {  
            int total = 0;  
            for (int i=0; i<b.length; i++)  
                total += b[i];  
            System.out.println(total);  
        }  
    }  
}
```

(10 marks)

Total (25 marks)

QUESTION 4

- (a) Rewrite the following code in more efficient way, (the loop can be replaced by a single MPI collective communication call)

```
for (i=0; i < numProcs; i++) { // numProcs is the number of processors  
MPI_Reduce(b, c, n, MPI_INT, MPI_SUM, i, MPI_COMM_WORLD);  
}
```

(5 marks)

- (b) Write a complete MPI program to find how many zeros there are in an integer array A[N] using a scatter/gather pattern. Define N as a constant with a value 10,000. Define the number of slaves as P and set to 10 slaves. Each slave is sent one group of N/P numbers from the array.

(14 marks)

- (c) Explain where does a hybrid OpenMP/MPI be most useful?

(6 marks)

Total (25 marks)

QUESTION 5

- (a) Write a sequential C program to compare two integer $N \times N$ arrays and report how many elements are bigger (i.e. if $A[i][j]$ is **Bigger** to $B[i][j]$, for all i and j) with a print statement. N is a defined constant set to 1000. Assume the arrays are initialized.

(5 marks)

- (b) Modify the code using OpenMP directives to parallelize the code using P threads, where P is also a defined constant and N is a multiple of P .

(10 marks)

- (c) Finally re-write the code to be an MPI program with P processes.

(10 marks)

Total (25 marks)