DUBLIN INSTITUTE OF TECHNOLOGY

_____

DT228 BSc. (Honours) Degree in Computer Science

Year 4

DT282 BSc. (Honours) Degree in Computer Science (International)

Year 4

DT211C BSc. (Honours) Degree in Computer Science (Infrastructure)

Year 4

_____

WINTER EXAMINATIONS 2018/2019

_____

Distributed Systems [CMPU4021]

Ms. E. Hatunic-Webster
Dr. D. Lillis
Mr. P. Clarke – DT228
Mr. P. Clarke - DT282
Dr. D. Malone – DT211C

.

Time allowed: 2 hours

Attempt **3 questions**
All questions carry **equal** marks
One complimentary mark is available

1. (a) Discuss the difference between *peer-to-peer* and *client-server* software architectures and give examples of systems they are suitable for.

(8 marks)

(b) The code shown below is a partial implementation of a server that takes a message from the user and sends it to *a group* of clients.
Complete the Java implementation of this code, and write a client that can receive the message. Note that several client processes should be able to run concurrently, with *all* of them receiving the message.

```
import java.net.*;
import java.io.*;
public class  Q1b{

 public static void main(String[] args) {

   try {
      int port = 22211;
      BufferedReader inputStr = new BufferedReader(new
                                          InputStreamReader(System.in));
      DatagramSocket socket = new DatagramSocket();

      InetAddress group = // WRITE THE MISSING CODE

      while(true) {
           System.out.print("Enter message :/> ");
           String message = inputStr.readLine();
           socket.send(new DatagramPacket(
                              message.getBytes(),
                              message.length(),
                              group, port));}
      } catch (Exception e) {}
   }
}
```

(12 marks)

(c) Describe the main usage of *indirect communication* and discuss *group* communication as an example of an indirect communication paradigm.

(13 marks)

2. (a) Describe *serial equivalence* in terms of concurrent transactions.

(8 marks)

(b) Explain the term *transparency* in relation to distributed systems, and using as examples *four* different types of transparency, show how they *are* or *are not* provided for by some distributed applications.

(12 marks)

(c) Consider a system offering a *football-score-service* that provides scores to client applications in real time, as scores change.

Present a design for a distributed object system for such a service and illustrate the role of remote objects and proxy objects. Evaluate the strengths and weaknesses of your design.

(13 marks)

3. (a) Describe the states that a *thread* can be in while running inside a process in a Java Virtual Machine (JVM).

(8 marks)

(b) Write the code for the multi-threaded Java program described below.

- The main thread starts off three threads to run concurrently.
- Each thread is responsible for doing the following inside its `run()` method:
    - Adding up all the numbers from 1 to some specific number.
    - Printing out the result of the addition.
    - The first thread must add all numbers from 1 to 500; the second must add all numbers from 1 to 50; the third must add all numbers from 1 to 5.

You must ensure that while the three threads run concurrently, the first thread MUST be the first to finish and the second thread MUST be the second to finish i.e. the threads must wait for each other.

(12 marks)

(c) Compare and contrast *web services* and the *distributed object model*. Use examples where necessary.

(13 marks)

4. (a) A distributed World Health Organisation (WHO) statistics application has statistic objects available to connecting clients. These remote objects allow a client to get:

- The main language of a country
- The population of a country
- The capital city of a country

Define the *interface* to the WHO statistics service in *Java Remote Method Invocation (RMI).*

(8 marks)

(b) With the help of a diagram and examples, explain the Message Oriented Middleware (MOM) paradigm and discuss its advantages and disadvantages.

(12 marks)

(c) Discuss the *three* different types of RMI *invocation semantics*.

(13 marks)