Q1) This question concerns concurrent programming in Java.

```
01  class Counter {
02    private volatile int c = 0;
03    public void increment() {
04      c++;
05    }
06    public void decrement() {
07      c--;
08    }
09    public int value() {
10      return c;
11    }
12  }
```

Consider the Counter class shown above. Consider to what extent an object of this class would be safe to use by multiple threads concurrently, by answering the following questions:

a) What problems might occur if multiple threads invoke the methods belonging to the class concurrently? (6 marks)

b) What is the effect of the *volatile* keyword as used in this class and how would it affect multiple threads invoking the methods belonging to the class concurrently? (6 marks)

c) On the basis of your answers to parts (a) and (b) describe features of multiple threads that could safely invoke this code concurrently, and on the other hand, features of multiple threads where invoking the code concurrently would be unsafe. (7 marks)

d) Propose a solution to make the class safe to use concurrently in any way, specifying the exact changes that would need to be made to the source code including specific line numbers for the changes. (6 marks)

(25 marks in total)

Q2) This question concerns parallel programming in Java.

Consider the following Java code:

```
01  public class Foo {
02    public static BigInteger fib( int index) {
03      BigInteger a = BigInteger.ZERO;
04      BigInteger b = BigInteger.ONE;
05      BigInteger c = a.add( b);
06
07      for( int i = 0; i < index; i++) {
08        a = b; b = c;
09        c = a.add( b);
10      }
11
12      return c;
13    }
14
15    public static void doCalculateFibs( int[] input,
16                                        BigInteger[] output){
17      for( int i = 0; i < input.length; i++) {
18        output[i] = fib( input[i]);
19      }
20    }
21  }
```

The function ***fib*** calculates the *n*th Fibonacci number where *n* is specified by the ***index*** parameter. The result is returned in a Java BigInteger object that can represent arbitrarily large numbers.

The function ***doCalculateFibs*** takes are array ***input*** that contains index values to compute Fibonacci numbers for, and an array ***output*** of the same size, to receive the Fibonacci numbers in. It iterates across these arrays using the ***fib*** method to compute the Fibonacci numbers.

a) Analyse this code and identify what parts *are* and *are not* suitable for parallelization, explaining why in each case. (5 marks)

The **RecursiveAction** class provided by the java.util.concurrent package is intended to be used as a base class for the implementation of parallel jobs. To use it, the developer must subclass it and provide an implementation of the abstract ***public void compute()*** method. The implementation must *either* perform a part of the work the job requires *or* fragment the work and delegate it to two new instances of the same subclass. In the latter case the newly created objects are then scheduled for execution using the static ***invokeAll*** method that is available in **RecursiveAction**'s own scope. The implementation of the ***compute()*** method should contain the conditional logic to decide whether to perform the work or delegate it.

b) Give a full class definition for a subclass of **RecursiveAction** suitable for parallelizing the work done by ***doCalculateFibs*** method shown above. This should show the fields of the class, constructor and overridden ***compute()*** method. This method should not contain a direct copy of the code in *doCalculateFibs*, but it should invoke the ***fib(int index)*** method in the Foo class in order to do the work. Import statements need not be shown. (15 marks)

c) Explain the specific condition you chose to decide whether to perform the work or delegate it and comment on any fixed values you chose to use for this in relation to the likely size of the input array processed by ***compute()*** . (5 marks)

(25 marks in total)

Q3) This question concerns network programming in Java.

a) Describe what is meant by a ***client/server*** architecture. (5 marks)

The following code represents a simple implementation of a server in Java

```
01  public class Server {
02    private static void handle( Socket cs)
03        throws IOException {
04      DataInputStream is = new DataInputStream(
05          cs.getInputStream());
06      DataOutputStream os = new DataOutputStream(
07          cs.getOutputStream());
08      while( true) {
09        String dir = is.readUTF();
10        if ( !dir.equals( "")) {
11          listDir( os, dir);
12        } else {
13          cs.close();
14          break;
15        }
16      }
17    }
18    private static void listDir( DataOutputStream os,
19                          String dir) throws IOException {
20      File fdir = new File( dir);
21      if ( fdir.isDirectory()) {
22        String[] contents = fdir.list();
23        if ( contents != null) {
24          for( final String s : contents) {
25            os.writeUTF( s);
26          }
27        }
28        os.writeUTF( "");
29      }
30    }
31    public static void main( final String[] args)
32          throws Exception {
33      ServerSocket server = new ServerSocket( 5566);
34      while( true) {
35        Socket clientSocket = server.accept();
36        handle( clientSocket);
37      }
38    }
39  }
```

b) Explain the meaning and purpose of each of lines 33, 35, 05 and 07. (5 marks)

c) Describe what happens when a client connects to this server. (10 marks)

This server has poor error handling, a highly insecure design, and lacks any mechanism for shutting down. It also cannot support multiple clients simultaneously.

d) Disregarding the other short-comings mentioned above, describe the steps that would need to be taken to enable it to handle multiple simultaneous clients. You should refer to the code but do not need to write any. (5 marks)

(25 marks in total)

Q4) This question concerns data formats for distributed systems.

Given the following Java class, write a JSON representation of the result

```
public class Record {
  private final int seq;
  private final double exp;
  private final Set<String> emails = new TreeSet<>();
  private final List<PhoneNum> phoneNumbers = new ArrayList< >();
  public Record( final int seq, final double exp) {
    this.seq = seq;
    this.exp = exp;
  }
  public void addEmail( String email) {
    this.emails.add( email);
  }
  public void addPhone( PhoneNum number) {
    this.phoneNumbers.add( number);
  }
  public enum PhoneType { PERSONAL, WORK };
  public static class PhoneNum {
    public final enum PhoneType type;
    public final String number;
    public PhoneNum( PhoneType type, String number) {
      this.type = type;
      this.number = number;
    }
  }
}
```

a) Write a sensible JSON representation of the record Object created by the following statements. (10 marks)

```
Record record = new Record( 2452363, 0.3456);
record.addEmail( "abc@gmail.com");
record.addPhone( new PhoneNum( PhoneType.WORK, "06791000111"));
record.addPhone( new PhoneNum( PhoneType.PERSONAL, "05622000111"));
```

An application that needed to maintain an online store consisting of collections of records defined by the Record class shown above could do so using a relational database with a server-side process providing access to it (for example a PHP script), alternatively it could make use of a generic cloud storage service based on storing JSON documents such as provided by Google's FireBase platform.

b) Outline the functionality of a PHP script (for example) providing access to a relational database would need to implement, including a brief description of the likely structure of the database. (8 marks)

c) Give reasons why an existing platform like FireBase is a more attractive option for many application developers. (7 marks)

(25 marks in total)

END OF EXAM PAPER