UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

Department of Computing

B. Sc. Examination 2015

IS52025A
Internet and Distributed Programming

Duration: 2 hours 15 minutes

Date and time:

There are five questions in this paper. You should answer no more than THREE questions. Full marks will be awarded for complete answers to a total of THREE questions. Each question carries 25 marks. The marks for each part of a question are indicated at the end of the part in [.] *brackets.*

There are 75 marks available on this paper.

THIS PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

Question 1

(a) (i) What are the disadvantages of single threaded server? [1]

(ii) Give a reason why `new Socket(x,y)` can throw an `Exception` [1]

(iii) Briefly explain the purpose of *Object Serialization* [1]

(iv) What does the `transient` keyword do? [1]

(v) Briefly explain two benefits of using threads over processes [2]

(vi) Briefly explain the purpose of the `Socket.accept()` method [2]

[8]

(b) Consider the following Java code, with 4 missing fragments:

```
import java.io.*;
import java.net.*;

Class client
{
     public static void main(String[] argv) throws Exception
       {
        Socket s = /* missing 1 */;
        OutputStreamWriter p =/* missing 2 */;
        InputStream i = /* missing 3 */;
        InputStream b = /* missing 4 */;
        int c;
        while(true)
          {
               c=b.read();
               p.write((char)c);
               p.flush();
               System.out.print((char)i.read());
          }
     }
}
```

Complete the missing fragments so that the above program acts as a client, which sends data one character at a time to a server running on the localhost listening on port 6000. It sends whatever is typed on the console one character at a time to the server and prints out on the console whatever characters it receives back from the server.

[8]

c) Write a complete single threaded server that listens on port 6000 for characters, converts them to upper case and sends them back to the client.

[9]

Question 2

a) Consider the following Java program with 5 missing fragments

```
import java.sql.*;

public class DBSelect {
  public static void main(String[] args) throws Exception
  {
     Class.forName("com.mysql.jdbc.Driver");
     Connection connect = DriverManager.getConnection(
               "jdbc:mysql://localhost:8889/java_demo"
               ,"user","password");
     String query = "SELECT * FROM messages WHERE user = ?";
     PreparedStatement st = /* missing 1 */
     /* missing fragment 2 */
     ResultSet rs = /* missing 3 */
     while (/* missing 4 */)
        {
          System.out.println(rs.getString("user") + ", "
                             + rs.getString("message"))
        }
     /* missing 5 */
     }
}
```

Complete the missing fragments so that the above program retrieves all the records from the database table `messages` where the `user` field has the value 'tom' and then prints out the database fields `user` and `message` (which are already defined columns in `messages`) on the terminal and then closes any opened resources.

[10]

b) Given the class Pair

```
class Pair

{

     String first;
     String second;
     Pair (String f,String s)
     {
        first=f;second=s;
     }
}
```

Write a method, makeSetFromTable which takes a ResultSet resulting from a query like;

```
ResultSet resultSet = st.executeQuery("SELECT *  from one");
```

and returns a `HashSet` of Pairs, each pair corresponding to a row of the table (which we can assume has two String fields).

[8]

c) Write a function makeTableFromSet which takes a HashSet of Pairs and inserts each pair one at a time into a table.

[7]

Question 3

a) Consider the following Java program

```
public class Greetings {

    static class Friend {
        private final String name;
        public Friend(String name) {
            this.name = name;
        }

        public String getName() {
            return this.name;
        }

        public void bow(Friend bower) {
            System.out.format("%s: %s"
                    + "  has bowed to me!%n",
                     this.name, bower.getName());
            bower.bowBack(this);
        }

        public void bowBack(Friend bower) {
            System.out.format("%s: %s"
                    + " has bowed back to me!%n",
                    this.name, bower.getName());
        }
    }

    public static void main(String[] args) {
        final Friend alphonse = new Friend("Alphonse");
        final Friend gaston = new Friend("Gaston");
        new Thread(new Runnable() {
            public void run() { alphonse.bow(gaston); }
        }).start();
        new Thread(new Runnable() {
            public void run() { gaston.bow(alphonse); }
        }).start();
    }
}
```

i. Explain what is the output when it is executed

ii. What would happen if we declared methods bow() and bowBack() as synchronized?

iii. What problem in concurrent programming is this an example of?

[9]

c) Write a complete multi-threaded Server that listens on port 5000 waiting for connections. Every time there is a new connection, it creates a new thread, which receives characters one at a time from the client each of which it sends back to the client.

[16]

Question 4

(a) Given the following class definition:

```
import java.io.*;

public class EngineStatus implements Serializable
{
     String name;
     int temperature;

     public EngineStatus (String n, int a)
     {
          temperature=a; name=n;
     }
     public String toString()
     {
        return name+" "+ temperature;
     }
}
```

Write a complete single-threaded client that repeatedly reads names and temperatures from the console, constructs `EngineStatus` objects from them, and sends these `EngineStatus` objects to a server listening on port 5000 on "localhost". It doesn't listen out for messages from the server.

[10]

b) Write a complete single-threaded 'Object' server that listens on port 5000 for Objects and prints them out on the console if they are `EngineStatus` Objects. It doesn't send messages back to its client.

[15]

Question 5

(a) Briefly describe the purpose of the following method:

```
static HashSet<String>  links (String url)
{
   HashSet<String> a= new HashSet<String>();
   try{
      org.jsoup.Connection z =Jsoup.connect(url);
      Document doc = z.get();
      Elements links = doc.select("a[href]")
      for (Element link : links) {
           a.add(link.attr("abs:href"));
      }
   }
   catch (Exception e)
   {
      System.out.println(e);
   }
   return a;
}
```

[9]

(b) Given a method `HashSet <String> links(String url)` write a method whose heading is

```
 static void Spider (String url, int n)
```

which finds and prints out *n* distinct links reachable from a URL given by the first parameter. It should find *all* links if there is less than *n* of them.

To do this, the spider should maintain two sets:

```
HashSet<String> alreadyVisited = new HashSet <String> ();
HashSet<String> toVisit = new HashSet <String> ();
```

[8]

(c) Rewrite your `Spider` method so that the spider stays within a particular domain. Write a `main` method which calls your `Spider` method. Very briefly explain how your `Spider` method works.

[8]