UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

B. Sc. Examination 2007

COMPUTER SCIENCE

IS53022AA(CIS337) Internet Programming in Java

Duration: 2 hours 15 minutes

Date and time:

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.

Electronic calculators are not allowed in this exam.

Your attention is drawn to the Appendix at the end of this paper.

THIS EXAMINATION PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

QUESTION 1

(a) Describe how threads are organised and managed by the Java Virtual Machine (JVM).

[5 Marks]

A thread is a separate line of execution within one JVM

Every thread has its own call stack

A run() method is placed on a new call stack and begins when the scheduler is ready

Only one thread of execution on a single-processor machine

If a thread becomes blocked for whatever reason, the JVM scheduler will continue execution with another thread (if there is another one)

(b) Write an account of the use of threads in server programming. You should compare threaded and non-threaded servers, and mention the general strengths and weaknesses of thread programming.

[10 Marks]

Each request is handled on a Java server by a different thread.

A server typically sets up a pool of 200 active threads which are allocated to clients in turn.

If the pool is busy, new requests might be queued, or even discarded

Threads are an efficient way of dealing with requests because each connection might take a while to service or become blocked

Threads occupy the same heap within the JVM i.e. threads share the same memory space

and this can make them difficult to programme due to concurrency problems.

However these can be solved by synchronizing methods

so that only one thread at a time has access.

Each process in a non-threaded serevr is a completely new program with its own memory

however processes are memory and CPU intensive.

(c) Write a simple server application, `SimpleServer`. This server should assign a new thread to each client request, and respond by printing the client address to the command terminal. Use an inner handler class for each client request.

[10 Marks]

```
// imports...
public class SimpleServer {

    boolean keepGoing = true;

    public SimpleServer() {

        try {
            ServerSocket serverSocket = new ServerSocket(7005);

             while (keepGoing) {

                Socket clientSocket = serverSocket.accept();
                Thread t = new Thread(new Handler(clientSocket));
                t.start();

            }
            serverSocket.close();
        } catch (IOException e) {
            System.out.println(e);
        }
    }

    public static void main(String[] args) {

        new SimpleServer();
    }

    class Handler implements Runnable {

        Socket socket;

        public Handler(Socket s) {
            socket = s;
        }

        public void run() {

            System.out.println("Connection from; " + socket);
        }
    }
}
```

QUESTION 2

(a) For security reasons, certain restrictions are placed on what Applets can and cannot do. What are these restrictions?

[5 Marks]

Award up to 5 for any of:

Applets CANNOT

access arbitrary memory address

access the local filesystem

launch other programs on the client

load native libraries or native method calls

use System.getProperty() to reveal information

manipulate any Thread that is not in its own ThreadGroup

open network connections to any host other than the originating host

listen to ports below 1024.

(b) Suppose that you have an applet, archived in `MyApplet.jar` that you wish to deploy in an HTML page. You wish the local browser to display an image `loading.gif` whilst the applet is loading. Write a few lines of HTML to show how you would insert this applet in your web page.

[5 Marks]

```
<applet code="MyApplet archive="MyApplet.jar" width = "500" height = "500">
<param name = "image" value="loaging.gif">
</applet>
```

(c) Applets have four lifecycle methods. What are these and what are their functions?

[5 Marks]

init() Initialise the applet (like a constructor)

start() Invoked when the page is loaded, reloaded or revisited by the Back button

stop() The user is leaving this page, or the applet is scrolled off-screen

destroy() The applet is being removed from memory. Here you can close files, network connections etc.

(d) Write an applet that plays a sound file whenever the page containing the applet is loaded by a browser. The sound file should stop playing if the browser leaves this page.

[10 Marks]

```
// imports
public class SimpleApplet extends Applet {
    AudioClip clip;

    public void init() {
        try {
            clip = getAudioClip(new URL(getCodeBase(), "Hello.au"));
        } catch (MalformedURLException e) {
            System.out.println(e);
        }
    }

    public void start() {
        if (clip != null)
            clip.play();
    }

    public void stop() {
        if (clip != null)
            clip.stop();
    }
}
```

QUESTION 3

(a) What is a servlet? How is it invoked, and what does it do?

[5 Marks]

A Servlet is a Java programm that runs on an HTTP web server.

When a client uses a web browser to interact with a web page, a request is sent back to the server.

If the request needs the help of a servlet, the web server runs the servlet code,

composes an MTML page and

sends it back to the client where it is displayed on the browser

(b) Itemise the steps in the servlet life-cycle

[3 Marks]

Firstly init() is called by the server. This initialises all the data and other objects required by the servlet.

Then, service() handles requests. The service() method accepts two parameters: a servlet request object and a servlet response object.

Finally destroy() is called when the server is closed down, in order to clean up.

(c) Write a simple servlet that responds in an appropriate way to a doGet message.

[7 Marks]

```
class HelloClient extends HttpServlet {

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {

        // must come first
        response.setContentType("text/html");
        PrintWriter out = response.getWriter(  );

        out.println(
            "<html><head><title>Hello World</title></head><body>"
            + "<h1> Hello Client </h1>"
            + "</body></html>" );
        out.close(  );
    }
}
```

(d) What is a Java Server Page (JSP), and how does it relate to servlets?

[4 Marks]

A servlet is a Java class that contains HTML in output statements

but a JSP is an HTML page that contains Java code.

JSP are dynamic web pages written in normal HTML, but with embedded Java that is trigerred by tags at runtime.

The main advantage is that it is easier to write Java inside HTML, than the write HTML in the servlet's print statements.

(e) Write a Java Server Page that displays the date and the time.

[6 Marks]

```
<HTML>
 <HEAD>
  <TITLE>Hello World</TITLE>
 </HEAD>
 <BODY>
  <H1>Hello World</H1>
  Today is: <%= new java.util.Date().toString() %>
 </BODY>
</HTML>
```

QUESTION 4

(a) What is RMI (remote method invocation)?

[3 Marks]

The JVM will only allow objects to communicate within the same heap.

RMI (remote method invocation) is designed to allow a call to a method on a different heap, running under a different instance of the JVM,

usually on a different computer.

(b) Explain, within RMI, the use of a client proxy.

[5 Marks]

The proxy

- also known as a stub -

is an object in the client's heap

that takes care of the networking

i.e. sockets, serialisation and streams

by packaging and sending calls to the server.

(c) Outline the steps for making the remote service that runs on an RMI server.

[10 Marks]

(Award up to 10 for any vlid point)

1. Make a remote interface, `MyRemote`, that defines the methods that a client can call remotely.

Both the stub and the service will implement this interface.

2. Make a remote implementation, `MyRemoteImpl.java`, of the remote methods defined in the interface.

This will do the actual computations needed by the client.

Register with the RMI system by including code in `MyRemoteImpl`:

`Naming.rebind("Remote Hello, this");`

3. Generate server and client stubs by running the rmic tool on the remote implementation.: `rmic MyRemoteImpl`

4. Start the RMI registry, using the command `rmiregistry`

5. Start the remote service `java MyRemoteImpl`

(d) How does the client get hold of the stub class and object?

[7 Marks]

(Award marks for valid points up to a total of 7)

The client gets the stub object with a call to Naming

```
MyRemote service = (MyRemote)Naming.lookup("rmi://127.0.0.1/Remote Hello")
```

where Remote Hello is the name used to register the service with the rmi registry on the server.

The client must have the stub class at the time of the lookup or else the de-serialization will not work.

Often the stub is hand-delivered,

but you can use dynamic class downloading.

The stub object contains a URL that tells the RMI system where to get the class,

using an HTTP Get to retrieve the class file from a web server

QUESTION 5

(a) Write a few lines of code to show how, using the URL class, a web page can be downloaded into a Java programme.

[5 Marks]

```
try{

   URL u = new URL("http://www.gold.ac.uk");

   Reader r = new BufferedReader(new InputStreamReader(u.openStream()));
   int c = 0;
   while ((c = r.read()) != -1) {
      ...
   }
}
catch(MalformedURLException e){System.out.println(e);}
catch(IOException e){System.out.println(e));}
```

(b) A web spider finds a new site by looking for possible URLs on a given page. Suppose that you have access to a method `public static ArrayList getTags(URL u)` that returns a list of HTML tags contained within the web page represented by `u`. You now wish to write a method that will extract, if possible, a host name from this tag.

Write this method, with signature `public static String extractURL(String tag)`, that will return any host name that might be contained within `tag`, otherwise returning `null`.

[10 Marks]

```
public static String extractURL(String tag) {

      String host = null;
      tag.trim();
      if (tag.startsWith("<a") || tag.startsWith("<A")) {

          for (int i = 0; i < tag.length();) {
              String sub = tag.substring(i);
              if (sub.startsWith("http://")
                      || sub.startsWith("HTTP://")) {

                  int j = 7;
                  for (; j < sub.length() && sub.charAt(j) != ’/’
                          && sub.charAt(j) != ’"’; ++j);

                  host = sub.substring(0, j);

                  i += 7;
```

```
                } else
                    ++i;
            }

        }

        return host;
    }
```

(c) Using the methods `extractURL` and `getTags`, which you do not need to write out again, write a simple web spider. This spider should store host names in a set, and print every new host discovered to the command line.

[10 Marks]

```
// imports...
public class SimpleSpider {

    TreeSet set = new TreeSet();

    public void search(String host) {

        try {
            URL u = new URL(host);

            set.add(host);

            ArrayList tags = TagStripper.getTags(u);
            for (int i = 0, n = tags.size(); i < n; ++i) {

                String nexthost = TagStripper.extractURL((String)tags.get(i));
                if (nexthost != null && !set.contains(nexthost)) {
                    System.out.println(nexthost);
                    search(nexthost);
                }
            }
        } catch (MalformedURLException e) {
            System.out.println(e);
        } catch (IOException e) {
            System.out.println(e);
        }
    }

    public static void main(String args[]) {

        SimpleSpider spider = new SimpleSpider();
        spider.search("http://www.gold.ac.uk");

    }

}
```

Appendix: Class summaries

```
class java.awt.Graphics
abstract void dispose()
abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)
abstract void drawLine(int x1, int y1, int x2, int y2)
abstract void drawOval(int x, int y, int width, int height)
abstract void drawString(String str, int x, int y)
abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle)
abstract void fillOval(int x, int y, int width, int height)
abstract void fillRect(int x, int y, int width, int height)
abstract void setColor(Color c)

class java.awt.color
static final black
static final BLACK
static final white
static final WHITE
static final red
static final RED
static final green
static final GREEN
static final blue
static final BLUE

class java.util.Date
public Date()
public boolean after(Date when)
public boolean equals(Object obj)
public boolean before(Date when)
public String toString();

class java.net.InetAddress
public static InetAddress getByName(String host) throws UnknownHostException
public String getHostName()
public String getHostAddress()
public static InetAddress getLocalHost() throws UnknownHostException

class java.net.URL
public URL(String spec) throws MalformedURLException
public final InputStream openStream() throws IOException
public String getHost()

class java.io.InputStream
public abstract int read() throws IOException
public int read(byte[] b) throws IOException
public int read(byte[] b, int off, int len) throws IOException
public void close() throws IOException

class java.io.OutputStream
public abstract int write() throws IOException
public int write(byte[] b) throws IOException
public int write(byte[] b, int off, int len) throws IOException
public int write(int b) throws IOException
public void close() throws IOException

java.net.Socket
public Socket(InetAddress address, int port) throws IOException
public InputStream getInputStream() throws IOException
public OutputStream getOutputStream() throws IOException
public void close() throws IOException

java.net.ServerSocket
public ServerSocket(int port) throws IOException
public void close() throws IOException
public Socket accept() throws IOException

java.applet.Applet
public URL getCodebase()
public AudioClip getAudioClip(URL u)

java.applet.AudioClip
pubic void play()
pubic void stop()
```