

UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

B. Sc. Examination 2005

COMPUTING AND INFORMATION SYSTEMS

IS52014A (CIS220)

**Graphical Object Oriented and Internet Programming in
Java**

Duration: 3 hours

Date and time:

This paper is in two parts, Part A and Part B. There are a total of three questions in each part. You should answer two questions from Part A and two questions from Part B. Your answers to Part A and Part B should be written in separate answer books.

Full marks will be awarded for complete answers to a total of four questions, two from Part A and two from Part B. 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 100 marks available on this paper.

Electronic calculators are not allowed in this exam.

Your attention is drawn to the list of method prototypes in the Appendix at the end of this paper.

Part A: answer TWO questions from this Part

Question 1

- a) "Design Patterns are generally not language specific." Discuss. [2 Marks]
- b) Explain each of:
- i. Inheritance for extension. [3 Marks]
 - ii. Inheritance for specialisation. [3 Marks]
 - iii. Inheritance for specification (abstraction). [3 Marks]
- c) Demonstrate your understanding of the Observer - Observable design pattern by explaining the following terms in plain English:
- i. Observer interface [2 Marks]
 - ii. Observable class [2 Marks]
 - iii. update method [2 Marks]
 - iv. addObserver method [2 Marks]
 - v. notifyObservers method [2 Marks]
 - vi. setChanged, hasChanged and clearChanged method [4 Marks]

Question 2

a) Consider the following three methods:

```
// The method min returns the smallest item in a given subarray
public static int min(int[] a, int lo, int hi){
    // REQUIRES:
    // EFFECTS:
    // MODIFIERS:
    int indx = lo;
    for (int i = lo+1; i <= hi; i++)
        if (a[i] < a[indx])
            indx = i;
    return indx;
}
// The method sort sorts an integer array in ascending order
public static void sort(int[] a){
    // REQUIRES:
    // EFFECTS:
    // MODIFIERS:
    for (int i = 0; i < a.length; i++){
        int indx = min(a, i, a.length-1);
        swap(a, i, indx);
    }
}
// the method swap is swapping two items in an given array
public static void swap(int[] a, int i, int j){
    // REQUIRES:
    // EFFECTS:
    // MODIFIERS:
}
```

i. Write the REQUIRES, EFFECTS and MODIFIERS clauses for these methods

[9 Marks]

ii. Complete the swap procedure so that method sort sorts a given integer array in ascending order.

[6 Marks]

b) Program the recursive method called power.

```
public static int power(int a, int b)
    // Requires: b non-negative
    // Effects: Returns a raised to the power of b
```

[6 Marks]

- c) Explain what it means when a method or a data field is "static" in java.
[4 Marks]

Question 3

- a) Explain the use of Frameworks in Java, your answer should include at least one example:

[5 Marks]

- b) Implement a class that has the following behaviour (you may solve all sub-questions within one class):

- i. Program a small class displaying a rectangle in the center of a 400x400 JFrame.

[6 Marks]

- ii. Change the size of the rectangle to 50x60 and fill it with the colour red.

[6 Marks]

- iii. Add the following functionality: When the mouse is moved within the rectangle's boundaries the color of the rectangle should change to green and when moved out again it should be changed back to red.

[8 Marks]

Part B: answer TWO questions from this Part.

(This Part ends with an Appendix of method prototypes.)

Question 4

(a) Write a method with the prototype

```
public static String toDottedQuad( byte[] address )
```

which takes an array of four signed bytes and returns a dotted quad IP address in the form "12.34.56.78". *Hint:* the following statement converts signed bytes to unsigned bytes

```
int unsignedByte  
    = signedByte < 0 ? signedByte + 256 : signedByte
```

[5 marks]

(b) Write a method with the prototype

```
public static String getUserInput()
```

which reads in a line from the standard input.

[5 marks]

(c) Write a method with the prototype

```
public static String lookup( String query )
```

which EITHER returns an IP address corresponding to the hostname `query` OR returns the corresponding IP address if `query` is an IP address. If `query` is not a valid IP address or hostname, the method should return an appropriate error message. Your method should make a call `toDottedQuad(byte[] address)`; you do not need to write this method out again.

[7 marks]

(d) Write a method with the prototype

```
public static void main(String[] args)
```

which implements the unix utility *nslookup*. You should make full use of the methods from parts (b)-(d) and you do not need to write them out again. *nslookup* utility has two modes: interactive and command line. In command line mode, *nslookup* will print the host names or IP addresses of IP addresses or hostnames entered at the command line. The utility enters interactive mode if no host name or IP address is given; it will then continue to print the corresponding IP address and hostname to hostnames and IP addresses that have been entered at the standard input. An example of three sessions with **nslookup** is given below.

[8 marks]

```
>java nslookup www.gold.ac.uk
158.223.1.1
>
```

```
>java nslookup 158.223.1.1
scorpio.gold.ac.uk
>
```

```
>java nslookup
Enter names and IP addresses. Enter exit to quit.
scorpio.gold.ac.uk
158.223.1.1
127.0.0.1
localhost
exit
>
```

Question 5

(a) Explain how a non-threaded server might handle incoming requests. [4 marks]

(b) Explain how a threaded server (for example, written in Java) might handle incoming requests. [4 marks]

(c) Discuss the relative merits of threaded and non-threaded servers. [5 marks]

(d) Explain how threads are implemented in Java. You should use code samples to illustrate your answer. [12 marks]

Question 6

(a) Someone in your development team has handed you `UDPDiscardServer.java`, an uncommented Java program. Your job is to include comments at appropriate places. The source code for this program is printed on the following page.

[19 marks in total]

```

// comment 1 [6 marks]
public class UDPDiscardServer {

    // comment 2 [1 mark]
    public final static int DEFAULT_PORT = 9;

    // comment 3 [1 mark]
    public final static int MAX_PACKET_SIZE = 65507;

    public static void main(String[] args) {

        // comment 4 [1 mark]
        int port = DEFAULT_PORT;
        byte[] buffer = new byte[MAX_PACKET_SIZE];

        // comment 5 [1 mark]
        if ( args.length > 0 )
            port = Integer.parseInt ( args[0] );

        // comment 6 [1 mark]
        DatagramSocket server =
            new DatagramSocket(port);

        // comment 7 [1 mark]
        DatagramPacket packet =
            new DatagramPacket(buffer, buffer.length);

        // comment 8 [1 mark]
        while (true) {

            // comment 9 [2 marks]
            server.receive(packet);

            // comment 10 [2 marks]
            String s = new String(
                packet.getData(), 0,
                packet.getLength() );

            // comment 11 [2 marks]
            System.out.println( packet.getAddress() +
                " at port " + packet.getPort()+ " says " + s);
        }
    }
}

```

(b) Suggest an improvement that you would make to this code.

[6 marks]

APPENDIX

java.net.InetAddress

```
public String getHostName()
public byte[] getAddress()
public String getHostAddress()
public static InetAddress getByName(String host)
    throws UnknownHostException
public static InetAddress[] getAllByName(String host)
    throws UnknownHostException
public static InetAddress getByAddress(byte[] addr)
    throws UnknownHostException
public static InetAddress getLocalHost()
    throws UnknownHostException
```

java.net.Socket

```
public Socket(String host, int port)
    throws UnknownHostException, IOException
public Socket(InetAddress address, int port) throws IOException
public InetAddress getInetAddress()
public InetAddress getLocalAddress()
public int getPort()
public int getLocalPort()
public SocketAddress getRemoteSocketAddress()
public SocketAddress getLocalSocketAddress()
public InputStream getInputStream() throws IOException
public OutputStream getOutputStream() throws IOException
public void close() throws IOException
public boolean isConnected()
public boolean isClosed()
```

java.net.ServerSocket

```
public ServerSocket() throws IOException
public ServerSocket(int port) throws IOException
public InetAddress getInetAddress()
public int getLocalPort()
public SocketAddress getLocalSocketAddress()
public Socket accept() throws IOException
public void close() throws IOException
public boolean isBound()
public boolean isClosed()
```