



EXAMINATIONS – 2014  
TRIMESTER 1

COMP 112  
INTRODUCTION TO  
COMPUTER SCIENCE

Time Allowed: THREE HOURS \*\*\*\*\* WITH SOLUTIONS \*\*\*\*\*

**Instructions:** Closed Book  
Attempt ALL Questions.

The exam will be marked out of 180 marks.

Silent non-programmable calculators or silent programmable calculators with their memories cleared are permitted.

Printed foreign language dictionaries are permitted.

Java Documentation will be provided with the exam script

No other material is permitted.

Answer in the appropriate boxes if possible — if you write your answer elsewhere, make it clear where your answer can be found.

There are spare pages for your working and your answers in this exam, but you may ask for additional paper if you need it.

## Questions

	Marks
1. Defining a Class	[16]
2. Files	[18]
3. Event driven input	[22]
4. ArrayLists of Objects	[34]
5. Aligning Shapes	[23]
6. 2D Arrays and images	[24]
7. Networking and IRC	[25]
8. AI and Min-Max search	[18]

## Question 1. Defining a Class

[16 marks]

Complete the `Package` class on the facing page which stores information about packages being delivered by a Courier Company.

A `Package` object should contain fields to store:

- `senderAddress`, which contains the address of the sender
- `deliveryAddress`, which contains the address to deliver to
- `IDCode`, which contains a unique integer identifier for the package.
- `priority`, which is a numeric code for how urgent the package is (3 is the default priority; 1 is the highest priority.)
- `handovers`, which is a list of `Handover` objects. Each `Handover` object records the details of a package moving to another stage of the delivery process.

`Package` should have a constructor that takes two `String` parameters for the addresses, and stores them in the relevant fields. It should also assign a unique `IDCode`, using a static field to keep track of the next ID to assign.

`Package` should have four methods:

- `hasID(int id)` which returns true when the package has the given `IDCode`.
- `getPriority()` which returns the priority of the package
- `setPriority(int p)` which sets the priority of the package.
- `addHandover(Handover h)` which adds a `Handover` record to the package

The header of the constructor and one of the methods are given.

## (Question 1 continued)

```
class Package{  
  
    private String senderAddress;  
    private String deliveryAddress;  
    private int IDCode;  
    private int priority = 3;  
    private ArrayList<Handover> handovers = new ArrayList<Handover>();  
  
    private static int NextID = 1000;  
  
    public Package(String from, String to){  
  
        senderAddress = from;  
        deliveryAddress = to;  
        IDCode = NextID++;  
    }  
  
    public boolean hasID(int id){  
        return IDCode == id;  
    }  
    public int getPriority (){  
        return priority ;  
    }  
    public void setPriority ( int p){  
        priority = p;  
    }  
  
    public void addHandover(Handover h){  
        handovers.add(h);  
    }  
  
}
```

## Question 2. Files

[18 marks]

The courier company has a log of the number of deliveries per day to each suburb for the last month. They need a program that will do some analysis of the data. Each line of the log file contains a day of the month, a number of deliveries, and the suburb the deliveries were made to.

An example of the first few lines of a log file are at the top of the facing page.

(a) [8 marks] Complete the following `suburbFraction` method, whose parameters are the name of a suburb and the name of the log file. `suburbFraction` should read the file, count the total deliveries to the given suburb and the total deliveries to all suburbs, and print out the percentage of all deliveries that were made to the suburb.

For example, on the data above, `suburbFraction("Karori");` should print

```
Karori: 25%
```

given that 20 of the 80 deliveries were to Karori.

```
public void suburbFraction(String suburb, String filename){
    try{
        int count=0;
        int suburbCount = 0;
        Scanner scan = new Scanner(new File(filename));
        while (scan.hasNext()){
            int day = scan.nextInt ();
            int delivs = scan.nextInt ();
            String sub = scan.nextLine().trim ();
            count += delivs;
            if (suburb.equals(sub)){
                suburbCount += delivs;
            }
        }
        scan.close();
        UI.printf ("%s: %d%%\n", suburb, suburbCount*100/count);
    }catch(IOException e){UI.println("file reading failed "+e);}
}
```

**(Question 2 continued)**

First lines of example log file:

```

1 10 Johnsonville
1 18 Mount Cook
1 12 Kelburn
1 15 Karori
2 20 Crofton Downs
2 5 Karori

```

**(b)** [10 marks] Complete the following `highestDelivery` method which will read the file and print out the day and suburb that had the highest number of deliveries to a single suburb.

For example, with the data above, `highestDelivery` should print

```
Max on day 2: Crofton Downs (20 deliveries)
```

```

public void highestDelivery(String filename){
    try{
        int maxDay=0;
        int maxDelivs = 0;
        String maxSuburb = "";
        Scanner scan = new Scanner(new File(filename));
        while (scan.hasNext()){
            int day = scan.nextInt ();
            int deliveries = scan.nextInt ();
            String suburb = scan.nextLine().trim ();
            if ( deliveries > maxDelivs){
                maxDelivs = deliveries;
                maxDay = day;
                maxSuburb = suburb;
            }
        }
        scan.close();
        UI.printf ("Max on day %d: %s (%d deliveries)%n",
            maxDay, maxSuburb, maxDelivs);
    }
    catch(IOException e){UI.println("File reading failed "+e);}
}
catch(IOException e){UI.println("File reading failed "+e);}
}

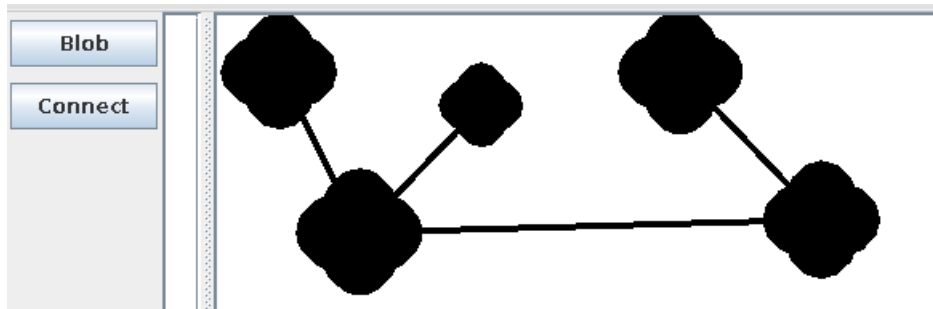
```

### Question 3. Event Driven Input

[22 marks]

Complete the ConnectBlobs program on the next **two** pages so that it allows the user to draw diagrams consisting of blobs connected by straight lines between the centers of the blobs. The program does not allow the user to edit the diagram.

The figure shows an example of a diagram that could be drawn.



The program should have two buttons:

- “Blob”: After clicking the “Blob” button, clicking the mouse on the graphics pane will create (and draw) a Blob at the mouse point. The program will remember the Blob in the list of Blobs.
- “Connect”: After clicking the “Connect” button, dragging the mouse from anywhere on one blob to anywhere on another Blob will draw a line between the centers of the blobs. If the mouse wasn’t pressed on a Blob or wasn’t released on a Blob, then there will be no line.

The program has a field, `blobs`, to store the list of all the Blobs that have been drawn. It does not need to store the lines.

The program has a `findBlob(double x, double y)` method that searches the list of Blobs for the Blob that the point  $(x, y)$  is on, and returns the Blob, or null if there is no such Blob.

The program should use the following Blob class:

Documentation of Blob **class**:

```
public Blob(double x, double y)
    // Constructor: creates a new Blob of a random size centered at the point (x, y).
    // Also draws the Blob on the graphics pane.

public boolean on(double x, double y)
    // Returns true if the point (x,y) is on top of the Blob.

public double getX()
public double getY()
    // Return the x and y coordinates of the center of the Blob
```

Hints:

- Lines can be drawn on top of the Blobs because they are all black.
- You may assume the user never draws one blob on top of another.

## (Question 3 continued)

```

public class ConnectBlobs implements UIButtonListener, UIMouseListener{
    private ArrayList<Blob> blobs = new ArrayList<Blob>();

    private boolean connecting = false;
    private Blob startBlob;

    public ConnectBlobs(){

        UI.setMouseListener(this);
        UI.addButton("Blob", this);
        UI.addButton("Connect ", this);

    }
    /** Returns the Blob that the point (x, y) is on, or null if no such Blob */
    public Blob findBlob(double x, double y){

        for (Blob blob : blobs){
            if (blob.on(x,y)){
                return blob;
            }
        }
        return null;

    }

    /** Respond to a button */
    public void buttonPerformed(String button){
        connecting = button.equals("Connect ");
// OR
        if (button.equals("Blob")) {
            connecting = false;
        }
        else if (button.equals("Connect ")) {
            connecting = true;
        }
    }

    [ mousePerformed method on next page!! ]

```

**SPARE PAGE FOR EXTRA ANSWERS**

Cross out rough working that you do not want marked.  
Specify the question number for work that you do want marked.



**(Question 3 continued)**

ConnectBlobs program continued:

```
/** Respond to the mouse */
public void mousePerformed(String mouseAction, double x, double y) {
    if (mouseAction.equals("pressed")){
        startBlob = findBlob(x, y);
    }
    if (mouseAction.equals("released")){
        if (connecting){
            Blob endBlob = findBlob(x, y);
            if (startBlob!=null && endBlob!=null){
                UI.drawLine(b1.getX(), b1.getY(), b2.getX(), b2.getY());
            }
        }
        else {
            blobs.add(new Blob(x, y));
        }
    }
}
```

}

## Question 4. ArrayLists of Objects

[34 marks]

This question concerns a `CourierManager` program to keep track of all the packages that a courier company is currently delivering. You are to complete four methods for this class.

The `CourierManager` class has a field that contains a list of `Package`s:

```
private ArrayList <Package> packages = new ArrayList<Package>();
```

The documentation of `Package` class is shown below.

(This question is independent of your answer to question 1, which addressed the implementation of `Package`).

---

### Documentation of the `Package` class

`Package`s have sender and delivery addresses, and a priority that is a positive integer. The **default** priority is 3; 1 is the highest priority.

Constructor:

```
Package(String from, String to){  
    // Constructor is passed the sender and delivery addresses of the package  
    // Sets priority 3  
    // Assigns a unique ID number.
```

Methods:

```
boolean hasID(int ID)  
    // returns true when the package has the given ID.  
  
int getPriority ()  
    // returns the priority of the package  
  
setPriority (int p)  
    // sets the priority of the package to p  
  
getHandovers()  
    // returns the list of Handover records  
  
toString ()  
    // returns description of a Package.
```

---

**(Question 4 continued)**

(a) [6 marks] Complete the following `accept` method of the `CourierManager` class that records a new package for delivery. It should ask the user for the sender address, the delivery address, and the priority, then create and add the new `Package` to the list of packages.

```
public void accept(){
    String from = UI.askString("from: ");
    String to = UI.askString("to: ");
    int pri = UI.askInt("priority: ");
    Package p = new Package(from, to);
    p.setPriority (pri);
    packages.add(p);
}
```

(b) [8 marks] Complete the following `deliver` method which is passed the ID of a package. The method should check whether the list contains a package with the specified ID. If so, the method should remove the package from the list; otherwise it should print a message such as "No package with ID 2441 in list".

```
public void deliver( int ID){
    for ( int i=0; i<packages.size(); i++){
        if (packages.get(i).hasID(ID)){
            packages.remove(i);
            return;
        }
    }
    UI.println ("Package "+ID+ " not found");
}
```

(Question 4 continued on next page)

**(Question 4 continued)**

(c) [8 marks] Complete the following `upgradePriorities` method which should upgrade the priority by 1 for every package that has more than 5 Handovers, unless its priority is already 1 (the highest priority).

```
public void upgradePriorities(){  
    for (Package p : packages){  
        if (p.getHandovers().size() > 5){  
            p.setPriority (p.getPriority ()- 1);  
        }  
    }  
}
```

**(Question 4 continued)**

**(d)** [12 marks] Complete the following `highestPriority` method which should return a list of all the highest priority packages in the `packages` field: if the highest priority package in `packages` has priority  $n$ , then the method will return a list of *all* the packages of priority  $n$ .

Hints:

- Remember that priority 1 is higher than priority 2.
- Repeatedly search through the list, starting with priority 1, and then repeating the search the list with decreasing priority if necessary, until some packages are found.

```
public ArrayList<Package> highestPriority(){
    ArrayList<Package> ans = new ArrayList<Package>();
    if (packages.isEmpty()) { return ans; }
    int priority = 1;
    while(ans.isEmpty()){
        for (Package p : packages){
            if (p.getPriority ()== priority ){
                ans.add(p);
            }
        }
        priority ++;
    }
    return ans;
}
```

## Question 5. Aligning Shapes

[23 marks]

The DiagramEditor program of assignment 7 allowed the user to construct diagrams using a variety of shapes. A desirable feature for the program would be the ability to align a list of shapes. You are to implement two of the many possible alignments.

(a) [10 marks] Complete the following `alignTop` method which is passed a list of Shapes, and changes the vertical position of all but the first shape so that their tops line up with the top of the first shape.

It should not change the horizontal position of the shapes, and it does not need to redraw the shapes.

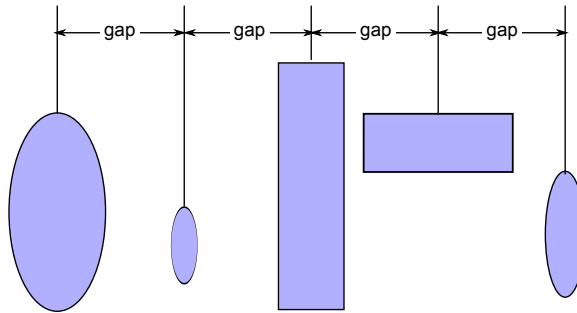
```
public void alignTop(ArrayList<Shape> list) {  
    if ( list .isEmpty()) { return; }  
    double top = list .get(0).getTop();  
    for(int i = 1; i < list .size (); i++){  
        Shape p = list .get(i);  
        p.setPos(p.getLeft (), top);  
    }  
}
```

Assume that the Shape interface is defined as follows:

```
public interface Shape {  
    public double getWidth();           // return the horizontal size of the shape  
    public double getHeight();         // return the vertical size of the shape  
    public double getLeft ();          // return the left size of the shape  
    public double getTop();            // return the top of the shape  
  
    public void setPos(double left, double top) // set the position of the shape  
  
    public void draw();                // redraw the shape  
    public boolean on(double x, double y); // return true if (x,y) is on the shape.  
}
```

**(Question 5 continued)**

**(b)** [13 marks] Complete the following `layoutHorizontally` method which is passed a list of `Shapes`, and moves them to distribute their centers evenly. The first and last shapes in the list should not be moved; the other shapes should be moved horizontally so that their horizontal centers are all separated by the same “gap”, as shown in the diagram, and the order of the items is the order they occur in the list. The gap can be computed from the number of shapes in the list and the horizontal difference between the centers of the first and last shapes. `layoutHorizontally` will do nothing if the list has fewer than three shapes.



```

public void layoutHorizontally ( ArrayList<Shape> list) {
    if ( list .size()<=2){ return; }
    double left = list .get(0).getLeft () + list .get(0).getWidth()/2;
    double right = list .get( list .size()-1).getLeft () + list .get( list .size()-1).getWidth()/2;
    double gap = (right - left )/( list .size()-1);
    for( int i = 1; i < list .size()-1; i++){
        Shape p = list .get(i);
        p.setPos(left+ gap*i - (p.getWidth()/2), p.getTop());
    }
}

```

## Question 6. 2D arrays and images

[24 marks]

Suppose an `ImageProcessor` program contains an `image` field which contains the `Color` values of the pixels in an image:

```
private Color [ ][ ] image;
```

(a) [12 marks] Complete the following `stretch` method which will stretch out the top half of the image to fill the whole image, replacing the bottom half of the image completely. For example, `stretch` would turn the image on the left into the image on the right.

Note: after calling `stretch`, the stretched image should be in the `image` field. You may use a temporary array if you wish, but it is not necessary.



```
public void stretch(){  
    int rows = image.length;  
    int cols = image[0].length;  
    for (int row=rows-1; row>0; row--){  
        for (int col=0; col<cols; col++){  
            image[row][col] = image[row/2][col];  
        }  
    }  
}
```



**(Question 6 continued)**

**(b)** [12 marks] Complete the following border method which will put a border of the specified width and colour around an image array, overwriting the current pixel values. For example, `border(10, Color.black)` would change the image on the left into the image on the right:



```

public void border(int bwidth, Color color){
    int rows = image.length;
    int cols = image[0].length;
    for ( int col=0; col<cols; col++){
        for ( int row=0; row<bwidth; row++){
            image[row][col] = color;
            image[rows-row-1][col] = color;
        }
    }
    for ( int row=0; row<rows; row++){
        for ( int col=0; col<bwidth; col++){
            image[row][col] = color;
            image[row][cols-col-1] = color;
        }
    }
}

```

## Question 7. Networking and the IRC client

[25 marks]

(a) [12 marks] The following code fragment is from the connect method in the code for an IRC client.

```
1  this.socket = new Socket (this.server, IRC_PORT);
2  this.serverIn = new Scanner (socket.getInputStream());
3  this.serverOut = new PrintStream (socket.getOutputStream());
4  this.send("NICK " + username);
5  this.send("USER " + username + " 0 unused :"+ realname);
6  while (this.serverIn.hasNextLine()) {
7      String line = this.serverIn.nextLine();
8      if (line.indexOf("004") >= 0) {return true;}
9      else if (line.indexOf("433") >= 0) {return false;}
    }
```

Briefly explain the role of each line in this piece of code:

1. open a new socket connected to the IRC server
2. create a Scanner on the input stream of the socket to be able to access the messages sent by the server
3. create a PrintStream on the output stream of the socket to be able to send messages to the server.
4. sending a NICK message to the server in order to log in.
5. sending a USER message to the server to complete the log in
- 6 and 7. loop that listens to all messages from the server
8. until it receives a message containing the code 004, indicating successful login or
9. it receives a message containing the code 433, indicating failure - nick name in use.

**(Question 7 continued)**

Answer to (a) continued:

**(b)** [6 marks] Lines 8 and 9 of the code fragment above use `line.indexOf`. Explain why this is a bad idea by saying why they might not work the way the programmer intended.

Note: a typical "433" message that the server might send is

```
:irc.ecs.vuw.ac.nz 433 * ChatBot :Nickname is already in use.
```

The code of a message should be in the second token of the message. If the message line contains 004 or 433 somewhere other than the second token of the message, and the program will think it has a success (or fail) message, but in fact the message might be something else completely.

**(Question 7 continued)**

**(c) [7 marks]** Explain why it was important in the IRC client to create a new thread to listen to messages from the server.

Because the server can send messages to the client independently of the client sending messages to the server, so there needs to be two separate threads to handle the two different sets of messages.

**Question 8. AI and min-max search**

[18 marks]

Given the following O's and X's board, and that it is O's turn to play, show how min-max search would determine which of the three possible moves is the best. Draw the the tree of possible future moves, and show how the value of each board state is propagated back up the tree, and explain the best move.

<table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px; text-align: center;">X</td><td style="width: 20px; height: 20px;"></td></tr> <tr><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">X</td></tr> <tr><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">X</td></tr> </table>				X		O	O	X		O	X																		
	X																												
O	O	X																											
	O	X																											
<p>O's turn /</p> <p>move 1</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px; text-align: center;">X</td><td style="width: 20px; height: 20px;"></td></tr> <tr><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">X</td></tr> <tr><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">X</td></tr> </table> <p>(LOSE)</p>		X		O	O	X		O	X	<p>move 2  </p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px; text-align: center;">X</td><td style="width: 20px; height: 20px;"></td></tr> <tr><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">X</td></tr> <tr><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">X</td></tr> </table> <p>(LOSE)</p>		X		O	O	X		O	X	<p>move 3 (*) \</p> <table border="1" style="margin: auto; border-collapse: collapse;"> <tr><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px; text-align: center;">X</td><td style="width: 20px; height: 20px; text-align: center;">O</td></tr> <tr><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">X</td></tr> <tr><td style="width: 20px; height: 20px;"></td><td style="width: 20px; height: 20px; text-align: center;">O</td><td style="width: 20px; height: 20px; text-align: center;">X</td></tr> </table> <p>(DRAW)</p>		X	O	O	O	X		O	X
	X																												
O	O	X																											
	O	X																											
	X																												
O	O	X																											
	O	X																											
	X	O																											
O	O	X																											
	O	X																											
<p>X's turn</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; border: 1px solid black; padding: 2px;">O   X   X</td> <td style="width: 20%; border: 1px solid black; padding: 2px;">O   X  </td> <td style="width: 20%; border: 1px solid black; padding: 2px;">  X   X</td> <td style="width: 20%; border: 1px solid black; padding: 2px;">X   X  </td> <td style="width: 20%; border: 1px solid black; padding: 2px;">X   X   O</td> <td style="width: 20%; border: 1px solid black; padding: 2px;">  X   O</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">  O   X</td> <td style="border: 1px solid black; padding: 2px;">X   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">  O   X</td> <td style="border: 1px solid black; padding: 2px;">X   O   X</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">LOSE</td> <td style="border: 1px solid black; padding: 2px;">(DRAW)</td> <td style="border: 1px solid black; padding: 2px;">LOSE</td> <td style="border: 1px solid black; padding: 2px;">(WIN)</td> <td style="border: 1px solid black; padding: 2px;">(WIN)</td> <td style="border: 1px solid black; padding: 2px;">(DRAW)</td> </tr> </table>			O   X   X	O   X	X   X	X   X	X   X   O	X   O	O   O   X	O   O   X	O   O   X	O   O   X	O   O   X	O   O   X	O   X	X   O   X	O   O   X	O   O   X	O   X	X   O   X	LOSE	(DRAW)	LOSE	(WIN)	(WIN)	(DRAW)			
O   X   X	O   X	X   X	X   X	X   X   O	X   O																								
O   O   X	O   O   X	O   O   X	O   O   X	O   O   X	O   O   X																								
O   X	X   O   X	O   O   X	O   O   X	O   X	X   O   X																								
LOSE	(DRAW)	LOSE	(WIN)	(WIN)	(DRAW)																								
<p>O's turn</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; border: 1px solid black; padding: 2px;">O   X   O</td> <td style="width: 20%; border: 1px solid black; padding: 2px;">X   X   O</td> <td style="width: 20%; border: 1px solid black; padding: 2px;">X   X   O</td> <td style="width: 20%; border: 1px solid black; padding: 2px;">O   X   O</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">X   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">O   O   X</td> <td style="border: 1px solid black; padding: 2px;">X   O   X</td> </tr> <tr> <td style="border: 1px solid black; padding: 2px;">DRAW</td> <td style="border: 1px solid black; padding: 2px;">WIN</td> <td style="border: 1px solid black; padding: 2px;">WIN</td> <td style="border: 1px solid black; padding: 2px;">DRAW</td> </tr> </table>			O   X   O	X   X   O	X   X   O	O   X   O	O   O   X	O   O   X	O   O   X	O   O   X	X   O   X	O   O   X	O   O   X	X   O   X	DRAW	WIN	WIN	DRAW											
O   X   O	X   X   O	X   X   O	O   X   O																										
O   O   X	O   O   X	O   O   X	O   O   X																										
X   O   X	O   O   X	O   O   X	X   O   X																										
DRAW	WIN	WIN	DRAW																										

**SPARE PAGE FOR EXTRA ANSWERS**

Cross out rough working that you do not want marked.  
Specify the question number for work that you do want marked.

\*\*\*\*\*