UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

B. Sc. Examination 2012

COMPUTER SCIENCE

IS52013A (CIS219 Resit) Graphical Object Oriented Programming in Java and Website Design

Duration: 3 **hours**

Date and time:

There are FIVE questions in this paper. Answer BOTH questions from Section A and TWO questions from Section B.

Full marks will be awarded for complete answers to a total of FOUR 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 100 marks available on this paper.

No calculators should be used.

Students are permitted to refer to the PHP reference sheet attached to the paper.

THIS EXAMINATION PAPER MUST NOT BE REMOVED FROM THE EXAMINATION ROOM

Section A

QUESTION 1

(a) Consider the following Java class:

```
interface ordering
{
    public abstract boolean compare(Object o1, Object o2);
}

class Date
{
 int day;
 int month;
 int year;
}
```

Provide an implementation of `ordering` which has a `compare` method which, when applied to two Datess, returns `true` if and only if the first Date is after than the second Date.

[8 Marks]

(b) Complete a generalised sorting method whose heading is

```
void sort( Object[ ] a, ordering o)
```

which sorts an array `a` of Objects relative to Ordering `o`. Hint:

```
void sort( Object[ ] a, ordering o)
{
         for (int i=0;i<a.length-1;i++)
         for (int j=i;j<a.length;j++)
         if (/*   */) // fill in this bit!
      {Object z=a[i];a[i]=a[j];aj]=temp;}
}
```

[4 Marks]

(c) Given an array of Dates `b`. Write a statement which calls the sort method above, which sorts b in time order according to going from earliest to latest.

[4 Marks]

.

(d) (i) Briefly describe the methods

```
V get(Object key)
Set<K>  keySet()
V  put(K key, V value)
```

of the Java interface `Map<K,V>`

(ii) Write a method

```
Date oldest (Map<Date,Integer> m)
```

Which returns a Date which is mapped to the highest value in a non empty map m.

[9 Marks]

QUESTION 2

(a) In the following partially completed Java program to allow two people to play n by n noughts and crosses, complete the ten missing portions of code.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class oxo1 implements ActionListener
{
 private JFrame window = new JFrame("Tic-Tac-Toe");
 private int N;
 private JButton board[][]; //represent the board as a two dim. array of JButtons
 private String letter;
 private int count = 0;

 boolean rowAllSame(int i, String s)
 // method to check whether row i has all elements equal to s
 {
   boolean same=true;
   for (int j=0;j<N && same;j++)
   {
     same = s.compareTo(board[i][j].getText())==0;
   }
   return same;
 }


 boolean columnAllSame(int i, String s)
 // method to check whether column i has all elements equal to s
 {
   boolean same=true;
   for (int j=0;j<N && same;j++)
   {
                                                    //!!!!!!!!!!!!!!!!(i) [1 mark]
   }
   return same;
 }




 boolean leftDiagAllSame(String s)
 // method to check whether the diagonal
```

```
// from top left to bottom right
// hass all elements equal to s
 {
  boolean same=true;
  for (int j=0;j<N && same;j++)
  {
    same = s.compareTo(board[j][j].getText())==0;
  }
    return same;
 }


 boolean rightDiagAllSame(String s)
// method to check whether the diagonal
// from top right to bottom left
// has all elements equal to s
 {
  boolean same=true;
  for (int j=0;j<N && same;j++)
  {
                                                    //!!!!!!!!!!!!!!!(ii)[1 mark]
  }
   return same;
 }


boolean win(String s)
{
 // checks whether s has won (s is eaither "X" or "O")
 // call methods above for checking rows,columns and diagonals

                                                    //!!!!!!!!!!!!!!!(iii)[3 marks]
}


void disableAll()
{
  //disable all the buttons
                                                    //!!!!!!!!!!!!!!!(iv) [2 marks]
}




public oxo1(int k) //constructor
```

```
{
 N=k;
 board = new JButton[N][N];
 window.setSize(50*N,50*N);
 window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 window.setLayout(new GridLayout(N,N));

 for (int i=0;i<N;i++)
   for (int j=0;j<N;j++)
   {
    // create the i,j button                    //!!!!!!!!!!!!!!!!(v)  [2 marks]
    // add the i,j button to the window         //!!!!!!!!!!!!!!!!(vi) [2 marks]
    // add the actionlistener to the i,j button //!!!!!!!!!!!!!!!!(vii)[2 marks]
  }
  window.setVisible(true);
}

public void actionPerformed(ActionEvent a)
{
  count++;
  if(count%2==0)letter = "X";
  else letter = "O";
    // set the pressed button to the letter     //!!!!!!!!!!!!!!!!(viii)[4 marks]
  b.setEnabled(false);

 if(/* letter has won */ )                      //!!!!!!!!!!!!!!!!(ix) [1 mark]
 {  JOptionPane.showMessageDialog(null, letter + " WINS!");
  disableAll();
 }
  else if (/*the board is full*/)               //!!!!!!!!!!!!!!!!(x)  [2 marks]
       JOptionPane.showMessageDialog(null, "DRAW");
 }

 public static void main(String[] args)
 {
    new oxo1(Integer.parseInt(args[0]));
 }
}
```

[**20 Marks**]

(b) Every colour is made up of a red, green and blue component, each of which is an integer between 0 and 255. In the following partially completed Java program the aim, is to produce a *Colour Mixer*. There are three sliders: one for red, one for green and one for blue. As we slide a slider to the right, the corresponding component of the colour being displayed is increased and conversely to the left, the colour is decreased. Complete the `stateChanged` method for the `ColourMixer` class.

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;

public class ColourMixer extends JFrame implements ChangeListener
{
 private JSlider[] bars = new JSlider[3];
 int rgb[]= new int[3];

    public ColourMixer()
    {
      setTitle("Colour Mixer"); setLayout(new FlowLayout());
      for (int i=0;i<3;i++)
      {
        bars[i]=new JSlider(0,255,JSlider.HORIZONTAL);
        bars[i].setValue(0);
        add(bars[i]);
        bars[i].addChangeListener(this);
        rgb[i]=0;
      }
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setSize(300,300); setLocation(300,300);
      getContentPane().setBackground(new Color(rgb[0],rgb[1],rgb[2]));
      setVisible(true);
     }

    public void stateChanged(ChangeEvent e)
    {
       //Complete this method!
    }

    public static void main(String[] args)
    {
      new ColourMixer();
    }
}
```

[5 Marks]

Part B. Answer TWO questions from this part.

Question 3.

This question concerns networks and communications protocols

a)

i. Explain the concept of **layering** in networked computing. How are **headers** used to implement layering in TCP/IP?

ii. Which TCP/IP layers do the following protocols belong to? Briefly state the functions defined for each protocol, for instance *DHCP: Internet layer protocol for automatically assigning IP addresses and other parameters to host computers*.

i. HTTP
ii. DNS
iii. ICMP

[7 marks]

b) In the TCP/IP model, which layer is primarily responsible for **reliable** (error-free) transmission? Explain in general terms how TCP/IP protocols ensure data is transmitted reliably, with reference to the **three-way handshake**. **[6 marks]**

c)

i. Explain what is meant by the **classful** system of IP addressing and routing in IPv4. Why this has been superseded by Classless Inter-Domain Routing (CIDR), and what are the implications for the design and contents of **routing tables**?

ii. How many host computers can be attached to a network with an IP prefix of /23?

iii. Explain how the following addresses are divided into network and host portions:

129.8.45.13/24
124.42.5.45/26
220.3.6.23/23

[12 marks]

Question 4.

This question concerns web programming and PHP scripting

a. Briefly explain the following:
   i. The differences between **GET** and **POST** methods in PHP scripts
   ii. The use of **cookies** and **session variables** in web applications.

[8 marks]

b. Appendix A shows a PHP script adapted from Larry Ullman's *PHP for the Web*. Explain what this script does and suggest suitable text for Messages 1, 2 and 3.

[9 marks]

c. The following is a (simplified) sequence of HTTP headers from a single transaction, captured using the **Live HTTP Headers** Firefox Add-on. Briefly explain the functions of each header and the meanings of these particular examples:

[8 marks]

```
GET /computing/ HTTP/1.1
Host: www.gold.ac.uk
User-Agent: Firefox/3.5.3
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-gb,en
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8
Connection: keep-alive
Referer: http://www.gold.ac.uk/

HTTP/1.1 200 OK
Date: Fri, 11 Feb 2012 15:28:31 GMT
Server: Apache/2.2.3 (Red Hat)
X-Powered-By: PHP/5.1.6
Cache-Control: private
Connection: close
Content-Type: text/html; charset=UTF-8
```

Question 5.

This question concerns security, accessibility and regulatory issues

a) Why is it important for website operators to ensure that their content can be accessed by people with disabilities? **[4 marks]**

b)

i. Explain in general terms how a **screen reader** renders a webpage. How will this influence the design and coding of a webpage that is likely to be accessed by partially sighted users?

ii. Appendix B shows an example of a web page coded in HTML. How could this page be modified to make it more accessible to users of screen readers?

[9 marks]

c) Explain what is meant by **cross-site scripting** and **SQL injection attacks**. How can you guard against these threats in a PHP script?

[6 marks]

d) Suppose you are operating a social networking website which allows users to upload their own content, including text, music and images. What steps would you take to meet your legal responsibilities regarding **copyright** and **intellectual property**? **[6 marks]**

Appendix A: script8.php

```
<?php
define('TITLE', 'Script8');
require('templates/header.html');
print '<h1>Heading</h1>';
if ( isset($_POST['done']) ) {
	if ( (!empty($_POST['email'])) && (!empty($_POST['password'])) ) {

		if ( $_POST['email'] == 'me@example.com') &&
		($_POST['password'] == 'testpass') ) {
			print '<p>Message 1</p>';
		} else {
			print '<p>Message 2</p>';
		}
	} else {
		print '<p>Message 3</p>';
	}
} else {
	print '<form action="script8.php" method="post">
	<p>Email Address: <input type="text" name="email" size="20" /></p>
	<p>Password: <input type="password" name="password" size="20"/></p>
	<p><input type="submit" name="submit" value="Log In!" /></p>
	<input type="hidden" name="done" value="true" />
	</form>';
}
require('templates/footer.html');
?>
```

Appendix B

```
<html>
<body>
<table height="100%" width="100%">
<tbody><tr><td align="center" valign="center" width="25%">
<h3>Stanley Fish</h3>
<p><font size="4"><b>Lecturer<b></b></b></font>
<br><font size="4"><b>Computer Science</b></font> </p><p>
</p>
</td>
<td colspan="4" rowspan="2" align="center" valign="center">
<h2>Teaching</h2>
<p><font size="4"><a href="Projects.html">BSc Projects</a></font>
<br><font size="4"><a href="Algorithms.html">Algorithms</a></font>
</p>
<p>
<font size="5">Research</font></p>
<p><font size="4">Software Agents</font>
<br><font size="4">Computational Creativity</font>
</p>
<p><a href="Publications.html">
<font face="Arial" size="5"><b>Publications</b></font></a></p></td></tr>
<tr><td align="center" valign="center">
<img src="mugshot.jpg" border="0" height="183" width="182"></td></tr>
 </tbody></table>
 </body>
 </html>
```