

UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

B. Sc. Examination 2010

COMPUTER SCIENCE

IS52014A (CIS220 Resit) Graphical Object Oriented and
Internet Programming in Java

Duration: 3 hours

Date and time:

Candidates should answer FOUR questions only.

Full marks will be awarded for complete answers to FOUR questions.

There are 100 marks available on this paper.

You must answer TWO questions from section A and TWO questions from section B.

Electronic calculators may NOT be used.

**THIS EXAMINATION PAPER MUST NOT BE REMOVED
FROM THE EXAMINATION ROOM**

SECTION A: Graphical programming
Answer TWO questions from this section.

Question 1

```
1 class PoorDogB
2 {
3     private int size;
4     private String name;
5
6     public int getSize()
7     {
8         return size;
9     }
10
11    public String getName()
12    {
13        return name;
14    }
15
16    public static void main (String[] args)
17    {
18        PoorDogB one = new PoorDogB();
19        System.out.println("Dog size is " + one.getSize());
20        System.out.println("Dog name is " + one.getName());
21    }
22 }
```

(a) Consider class *PoorDogB* above

- (i) Give the names of any instance variables in the class [1 mark]
- (ii) What will be the output of lines 19 and 20? [4 marks]
- (iii) Write a constructor for *PoorDogB* that allows the user to set the values of the instance variables [6 marks]
- (iv) If class *PoorDogB* is compiled with your new constructor added but no other changes, it will result in a compiler error. Give the current line number of the code that will cause the compiler error and explain why the class will not compile. [4 marks]

(b) How many constructors can an object have? Explain your answer with reference to overloading. [3 marks]

(c) Consider the class *Boo* below, and its subclass *SonOfBoo*

```
1 public class Boo
2 {
3     public String s;
4     public int i;
5
6     public Boo(int i)
7     {
8         this.i=i;
9     }
10
11    public Boo(String s)
12    {
13        this.s = s;
14    }
15
16    public Boo(String s, int i)
17    {
18        this.s = s;
19        this.i = i;
20    }
21 }
22
23 class SonOfBoo extends Boo
24 {
25     public SonOfBoo(String s, int i)
26     {
27         this(s);
28         super(i);
29     }
30
31    public SonOfBoo()
32    {
33        super("boo");
34    }
35
36    public SonOfBoo(int i)
37    {
38        super("Fred");
39    }
40
41    public SonOfBoo(String s)
42    {
43        super(42);
44    }
45
46    public SonOfBoo(String s, int i)
47    {
48        this(s, i, i);
49    }
50
```

```
51     public SonOfBoo(String a, String b, String c)
52     {
53         super(a,b);
54     }
55
56     public SonOfBoo(int i, int j)
57     {
58         super("man", j);
59     }
60 }
```

- (i) Explain the purpose of the call to *super()* in the constructors in *SonOfBoo*. [1 mark]
- (ii) *SonOfBoo* has 7 constructors, 3 of which will cause compiler errors. Identify the three constructors that will cause compiler errors by the line number of their signature. [3 marks]
- (iii) Explain briefly why each constructor that you identified in (ii) will cause a compiler error. [3 marks]

Question 2

(a) Consider the class *GC* below

```
public class GC
{
    public static GC doStuff()
    {
        GC newGC = new GC();
        doStuff2(newGC);
        return newGC;
    }

    public static void main(String[] args)
    {
        GC gc1;
        GC gc2 = new GC();
        GC gc3 = new GC();
        GC gc4 = gc3;
        gc1 = doStuff();
        A
        //some more code here
    }

    public static void doStuff2(GC copyGC)
    {
        GC localGC;
    }
}
```

- (i) Draw a memory diagram for class *GC*, showing what is on the stack and what is on the heap, by the time that point *A* in the code has been reached. [6 marks]
- (ii) Does your diagram show any unreachable objects? [1 mark]
- (iii) For each of the five cases below say whether or not it would, if inserted by itself into *GC* at point *A*, cause an object to become eligible for garbage collection? In each case give the line number and explain why or why not the code would make an object eligible for garbage collection if inserted into class *GC*. [10 marks]

```
1    gc1=gc4;
2    copyGC = null;
3    gc3 = gc2;
4    gc2 = null;
5    gc4 = null;
```

(b) Consider the class, Q2, below:

```
import java.util.*;

public class Q2
{
    public static ArrayList<String> list;

    public static void main(String[] args)
    {
        list = new ArrayList<String>();
        list.add(new String("Hello"));
        list.add(new String("Bongu"));
        list.add(new String("Drood"));
        list.add(new String("Salaam Aleikum"));
        list.add(new String("ni hao"));
        list.add(new String("Goodbye"));

        deleteString(list, "Goodbye");
        findString(list, "Goodbye");
        findString(list, "Drood");
    }

    public static void deleteString(ArrayList l, String s)
    {
        l.remove(s);
    }

    public static void findString(ArrayList l, String s)
    {
        int index = l.indexOf(s);
        if (index > -1) System.out.println(l.get(index));
        else System.out.println(s+" not found");
    }
}
```

- (i) The ArrayList in the above code has been parameterised. [2 marks]
Explain what parameterising means.
- (ii) What syntax is used to parameterize array lists? [2 marks]
- (iii) What will be the output of class Q2a when compiled and run? [4 marks]

Question 3

Consider the code below.

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

public class FirstGUI
{

    public static void main (String[] args)
    {
        FirstGUI gui = new FirstGUI();
        gui.go();
    }

    public void go()
    {
        JFrame frame = new JFrame();
        frame.setSize(500,500);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

- (a) Add a *JButton* to the *FirstGUI* class that initially displays the text "Question 3" and then when clicked displays "You clicked me!" [12 marks]
- (b) Explain how the *actionPerformed()* method gets called, and why it is that you do not have to explicitly call it in your code. [3 marks]
- (c) Consider the following:

```
public abstract class H
{
    abstract void hello(int n);
}

public interface G
{
    abstract void goodbye(int n);
}
```

Write a concrete class *C* that extends *H* and implements *G*. The method *hello(n)* should print out "hello" to the screen *n* times and the method *goodbye(n)* should print "goodbye" to the screen if *n* is less than 10. [10 marks]

SECTION B: Internet programming
Answer TWO questions from this section

Question 4

- (a) Write a program called *ColourCircle.java* to do the following:
- (i) Display a 400 x 400 *JFrame* labelled "Question four" [5 marks]
 - (ii) By writing an inner class of *ColourCircle* called *NewDrawPanel* that extends *JPanel* and overrides the *paintComponent()* method (or otherwise), draw a blue oval with the coordinates 125,125,150,150 [5 marks]
 - (iii) Using the *BorderLayout* manager add your blue oval to the CENTER region [5 marks]
 - (iv) Using the *BorderLayout* manager add a *JButton* to the SOUTH region that displays the text "Blue circle above" [5 marks]

You may answer questions (i)-(iv) in one class

- (b) What would be the output of the following code? [5 marks]

```
class exep
{
    public static void main(String [] args)
    {
        try
        {
            Integer.parseInt("aaa");
            System.out.println("hello");
        }

        catch (Exception e)
        {
            return;
        }

        finally
        {
            System.out.println("Let's finish");
        }
    }
}
```


Question 5

(a) Consider the class *PageSourceDisplayer* below

```
import java.io.BufferedReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.Reader;
import java.net.MalformedURLException;
import java.net.URL;

class PageSourceDisplayer
{
    public static String source;

    public static void main (String[] args)
    {
        StringBuffer buff = new StringBuffer();
        String s = "http://www.gold.ac.uk";
        try
        {
            URL u = new URL(s);
            Reader r = new BufferedReader(new
                InputStreamReader(u.openStream()));
            int c = 0;
            while ((c=r.read()) !=-1)
            {
                buff.append((char)c);
                source = buff.toString();
            }
            System.out.println(source);
        }

        catch(MalformedURLException e)
        {
            System.err.println(s+" is not a parseable URL");
        }

        catch(IOException e)
        {
            System.err.println(e);
        }
    }
}
// end main
}
// end class
```

- (i) Describe what the *PageSourceDisplayer* class will output when it is run on a computer connected to the Internet. [5 marks]
- (ii) If *PageSourceDisplayer* was run on a computer not connected to the internet what would be the output? [4 marks]

(b) Consider the class *MyRunnable*, below

```
public class MyRunnable implements Runnable
{
    public void run()
    {
        go();
    }

    public void go()
    {
        doMore();
    }

    public void doMore()
    {
        System.out.println("top o' the stack");
    }

    public static void main (String[] args)
    {
        Runnable theJob = new MyRunnable();
        Thread t = new Thread(theJob);
        t.start();
        System.out.println("back in main");
    }
}
```

When *MyRunnable* is compiled and run it will output either:

```
back in main
top o' the stack
```

or

```
top o' the stack
back in main
```

- (i) Why is it that the order of the output from *MyRunnable* can vary? [2 marks]
- (ii) Add some code to the class to make back in main always print before top o' the stack [6 marks]
- (c) A thread attempting to run a synchronized method must get the key first. If it is held by another object the thread has to wait until the other thread finishes and key is available. How does this give rise to the *thread deadlock* problem? [4 marks]
- (d) Why is thread deadlock a particularly serious problem in Java? [4 marks]

Question 6

- (a) Which of the following is saved when an object is *serialized*? [2 marks]
- (A) An object's state, given by its instance variables is saved. Any objects that are referenced by the instance variables, and in turn any further objects referenced by their instance variables etc are also saved.
 - (B) An object's static and instance variables are saved. Any objects that are referenced by the instance variables, and in turn any further objects referenced by their instance variables etc are also saved.
 - (C) The object's source code

- (b) An object whose superclass has implemented Serializable, but that has not itself implemented the interface is serializable. [2 marks]

True or false?

- (c) What will be the value of a transient variable when its containing object is deserialized? [2 marks]

- (d) Consider the following code:

```
import java.io.*;

public class Q6d implements Serializable
{
    private int x;
    private int y;

    public Q6d ()
    {
        x = 50;
        y = 90;
    }

    public static void main (String[] args)
    {
        String a = "hello";
        Q6d b = new Q6d();

        try
        {
            FileOutputStream f = new FileOutputStream("MyObj.ser");
            ObjectOutputStream os = new ObjectOutputStream(f);
            os.writeObject(a);
            os.writeObject(b);
            os.close();
        }
    }
}
```

```

        catch(Exception ex)
        {
            ex.printStackTrace();
        }
    }
}

```

(i) Explain briefly what the *try* block in the main method of the above class, *Q6d*, is doing to the String object and the *Q6d* object [4 marks]

(ii) Write some extra code to add to the main method of *Q6d*. Your code should read from the 'MyObj.ser' file any objects saved there, and cast them back to their original types. [6 marks]

(e) Consider the following method:

```

public void go()
{
    try
    {
        Socket s = new Socket("190.165.1.103", 4242);
        InputStreamReader i=new InputStreamReader(s.getInputStream());
        int c;
        while(true)
        {
            c=i.read();
            System.out.print((char)c);
        }
    }

    catch(IOException ex)
    {
        ex.printStackTrace();
    }
}

```

(i) The above method is listening for output from a machine. Which machine is it listening to? [2 marks]

(ii) Which port on the machine is it listening to? [2 marks]

(iii) What does it do with any data that it 'hears'? [3 marks]

(f) When writing new server programs we always choose a port number between 1024 and 65535. What stops us from using a port number between 0 and 1023? [2 mark]