

UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

B. Sc. Examination 2007

COMPUTER SCIENCE

IS52018A(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 Appendices at the end of this paper.

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

PART A

QUESTION 1

- (a) Write a Java class `SimpleGui`. This class should have an instance method `public void go()` and a static `main` method. You do not need to provide a constructor for this class. The `main` method should instantiate a `SimpleGui` object and call `go()` on this object. However the `go()` method should be left blank.

[5 Marks]

```
public class SimpleGui {
    public void go(){

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

- (b) Add code to `SimpleGui.java` so that the result of calling `go()` is to create and display a window of size 300 x 300. The programme should terminate when a user closes this window.

[5 Marks]

```
public void go(){
    JFrame frame = new JFrame();
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.setSize(300, 300);
    frame.setVisible(true);
}
```

- (c) Explain Java's event handling mechanism. You should consider the relationship between the event source (e.g. a button) and an object that should take some action as a result of this event occurring.

[10 Marks]

An application must listen for a GUI event in order to know when the user takes some action (such as clicking the button)

Listeners must be registered with an event source (e.g. button)

The listener interface provides a call-back mechanism

The interface defines a method that the event source will call when the event happens

Objects are registered for events with the source by calling the source's registration method

This is of the form `add<EventType>Listener`

For example, `addActionListener(this)` registers the object referenced by `this` for the ActionEvents fired by a button

The object must implement the interface's event-handling methods

The desired action is placed in the listener call-back method e.g. within `public void actionPerformed(ActionEvent e` for a button-clicked event

An event object is passed to the event-handler method - this object contains information about the type and the source of the event

- (d) Add a `JButton` to the window created in `SimpleGui`. This button should display the current date and time when a user clicks it.

[5 Marks]

```
// imports...
public class SimpleGui implements ActionListener{
    JButton button;

    public void go(){
        JFrame frame = new JFrame();
        button = new JButton();

        button.addActionListener(this);
        frame.getContentPane().add(button);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 300);
        frame.setVisible(true);
    }

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

    public void actionPerformed(ActionEvent arg0) {
        Date now = new Date();
        button.setText( now.toString() );
    }
}
```

QUESTION 2

- (a) Inheritance is a key idea in object oriented programming. In a few sentences, explain the essence of this idea.

[5 Marks]

(NB non-static methods only) A subclass inherits all the non-private methods and instance variables of its superclass

Inherited methods can be overridden

The subclass can add extra, more-specialised methods and instance variables

A superclass reference can point to a subclass object

In this case, if the subclass overrides a superclass method, the overridden version of the method is invoked (dynamic binding)

- (b) Under what circumstances should you use inheritance in your own coding? When should you guard against unnecessary use of inheritance?

[5 Marks]

Use inheritance when one class is more specific e.g. Dog IS-A animal

Use inheritance when you have some behaviour that should be shared amongst multiple classes of the same type.

Do not use just because you wish to re-use code from another class if the relationship between these classes violates the above two rules

Do not use if the subclass and superclass do not pass the IS-A test

Do not construct a deep inheritance tree because this separates code by too many steps. Keep the tree broad and shallow i.e. inheritance increases dependency and coupling

- (c) In general terms, what are the advantages of inheritance?

[5 Marks]

Common code is in one place

This aids maintenance and development because updates need only be made in one place

Inheritance defines a common protocol for a group of classes

This allows other developers to add subtypes to your system by ensuring that they conform to the supertype

Polymorphism: a supertype reference can point to a subtype object. This means that you don't have to change code when you introduce new subtypes

- (d) Appendix 1 shows a number of classes of a Jungle simulation. The class below, `JungleTestRun` has been written to test the simulation. What is the output of `JungleTestRun.main()`?

```
public class JungleTestRun {  
  
    public static void main(String[] args) {  
  
        CreepyCrawly bug = new GiantMillipede();  
        bug.scare();  
    }  
}
```

[5 Marks]

```
In Animal.Animal()  
In CreepyCrawly.CreepyCrawly()  
In GiantMillipede.GiantMillipede()  
Crawling up your leg!!  
In GiantMillipede.crawl()
```

- (e) Which (if any) of the following lines will compile if they are added to `JungleTestRun.main()`? Explain your answer.

```
Cat c = new Cat();  
Cat d = new Lion();
```

[5 Marks]

`Cat c = new Cat()` will not compile because `Cat` is an abstract class. (2 marks)

However `Cat d = new Lion()` WILL compile because `Lion` is concrete and you can have an abstract reference to a concrete class (3 marks)

QUESTION 3

- (a) The Java Virtual Machine (JVM) organises memory into two parts; the stack and the heap. What is the stack and how is it used by the JVM?

[5 Marks]

The stack is an ordered block of memory

The JVM pushes method invocations and values of local variables onto the stack

Technically this is called a stack frame

The method at the top of the stack is the currently executing method

When the method ends, the stack frame is removed and the method below resumes

- (b) What is the heap and how is it used by the JVM?

[5 Marks]

The heap has no defined order

All objects are placed in the heap

This is true if the reference to this object is local or if it is an instance variable

The heap is garbage collectible

this means that the garbage collector destroys any objects that are no longer referenced

- (c) Draw a stack-heap diagram to illustrate the relationships between variables and objects for the code below:

```
class DogPound{  
  
    public static void main(String[] args){  
  
        Dog dog1 = new Dog();  
        Dog dog2 = new Dog();  
        ...  
    }  
}
```

[5 Marks]

- (d) Draw a stack-heap diagram to illustrate the relationships between variables and objects just after line 3 for the code below. What will happen to the object referenced by `dog1` in line 1?

```
class DogPound{  
  
    public static void main(String[] args){  
  
        Dog dog1 = new Dog(); // line 1  
        Dog dog2 = new Dog(); // line 2  
        dog1 = dog2; // line 3  
        ...  
    }  
}
```

[5 Marks]

The Dog object in line 1 is eligible for garbage collection.

- (e) An object is eligible for garbage collection when its last live reference disappears. Demonstrate with code excerpts how this might happen.

[5 Marks]

This might happen in one of three ways

1. The reference is permanently out of scope

```
void go(){  
    Dog bongo = new Dog();  
} // end scope
```

2. The reference is re-assigned

```
Dog bongo = new Dog();  
bongo = new Dog(); // re-assign
```

3. The reference is set to null

```
Dog bongo = new Dog();  
bongo = null;
```

Award two marks for each circumstance and code snippet up to a total of 5

PART B

QUESTION 4

- (a) What is a thread?

[1 Marks]

A thread is a separate line of execution within one JVM

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

[4 Marks]

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)

- (c) Describe how threads are created and launched in the Java programming language.

[5 Marks]

You can implement the `Runnable` interface

by supplying a `public void run()` method.

To launch a new thread, pass a `Runnable` to the `Thread`'s constructor

and call `start()` on the `Thread` object

and this will place `run()` on a new stack

(Also award marks for extending `Thread`, and for showing how to use an adapter)

- (d) Illustrate your answer above by writing a Java programme to create and launch a thread.

[5 Marks]

```
class SayGoodbye implements Runnable {  
    public void run() {  
        System.out.println("Goodbye");  
    }  
  
    public static void main(String[] args) {  
        Thread t = new Thread(new SayGoodbye());  
        t.start();  
    }  
}
```

(e) Why are threads so important in server programming?

[5 Marks]

A server can only handle one client at a time

However a request might take a while to complete (for example due to a slow connection) rendering the server inactive

If each client is handled on a separate thread

the thread handling the slow request can become blocked

enabling the scheduler to choose another client-thread and the server can resume useful work

(f) The programme below shows a simple server application, `SimpleServer`. This server assigns a new thread to each client request, and responds by printing the client address to the command terminal. However the inner handler class has not been completed. Add code to `Handler` so that the server functions as described.

[5 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;  
        // ADD YOUR CODE HERE  
    }  
}
```

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

    ...

    class Handler implements Runnable {

        Socket socket;

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

        public void run() {

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

QUESTION 5

- (a) Write a few lines of Java code to show how a client might attempt to make a TCP connection to a server at 158.223.1.108 on port 5001.

[5 Marks]

```
try{
    Socket socket = new Socket("158.223.1.108", 5001);
    //...
    socket.close();
}
catch(IOException e){
    System.out.println(e);
}
```

- (b) Extend your code from the above answer so that the client can read textual messages sent from the server and print them to the command line

[5 Marks]

```
try{
    Socket socket = new Socket("158.223.1.108", 5001);

    // code for part (b)
    BufferedReader reader = new BufferedReader(
        new InputStreamReader(
            socket.getInputStream()));

    String line = null;
    while((line = reader.readLine()) != null){
        System.out.println(line);
    }
    reader.close();

    socket.close();

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

- (c) Write a few lines of code to demonstrate how a Java programme might read from the command line.

[5 Marks]

```

String line = null;
try {
    java.io.BufferedReader inputStream = new java.io.BufferedReader(new java.io.InputStreamReader(System.in))
    line = inputStream.readLine();
    inputStream.close();
} catch (java.io.IOException e) {
    System.err.println(e);
}

```

- (d) Write a few lines of code to show how a server can wait for a connection request on a designated port.

[5 Marks]

```

try{
    ServerSocket ss = new ServerSocket(5001);
    Socket sock = ss.accept();
    // deal with connection
    sock.close();
    ss.close();
}
catch(IOException e){
    System.out.println(e);
}

```

- (e) Explain how you could engineer a peer-to-peer chat application in Java.

[5 Marks]

The application would listen on a designated port using a ServerSocket object
 When accept() returns, open an input and an output stream on the socket
 These would preferentially be in separate threads
 Read and write to these streams from the terminal
 Close the streams, sockets and server socket when the conversation is over

QUESTION 6

- (a) URLs are represented in Java by the `URL` class. Show, in a few lines of code, how to instantiate a `URL` object for a given address.

[5 Marks]

```
try{
    URL u = new URL("http://www.gold.ac.uk");
    //...
}
catch(MalformedURLException e){
    System.out.println(e);
}
```

- (b) Show, in a few lines of code, how a `URL` object `u` can be used to download html source into a `String`.

[5 Marks]

```
try{
    Reader r = new BufferedReader(new InputStreamReader(u.openStream()));
    StringBuffer sb = new StringBuffer();
    int c = 0;
    while ((c = r.read()) != -1) {
        sb.add(c);
    }
    String thePage = sb.toString();
}
catch(IOException e){
    System.out.println(e);
}
```

- (c) Show, in a few lines of code, how a `URL` object `u` can be used to display a web page using Swing.

[5 Marks]

```
try {
    JEditorPane jep = new JEditorPane();
    jep.setEditable(false);
    jep.setPage(u);
    JFrame f = new JFrame(homePage);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.getContentPane().add(scrollPane);
    f.setSize(512, 512);
}
```

```

        f.show();
    }
    catch(IOException e){
        System.out.println(e);
    }
}

```

- (d) Extend your answer to the previous question by adding code that allows the user to click hyperlinks on the displayed page and therefore navigate to a new web page. (Hint: Write a LinkFollower class that implements the HyperlinkListener interface by supplying a public void hyperlinkUpdate(HyperlinkEvent e) method.)

[10 Marks]

```

/*
 * Display a web page on a swing JEditorPane
 */
package ipj;

import java.io.IOException;
import java.net.URL;

import javax.swing.JEditorPane;
import javax.swing.JFrame;
import javax.swing.JScrollPane;
import javax.swing.event.HyperlinkEvent;
import javax.swing.event.HyperlinkListener;

public class SimpleBrowser {

    public static class LinkFollower implements HyperlinkListener {

        private JEditorPane pane;

        public LinkFollower(JEditorPane jep) {
            pane = jep;
        }

        public void hyperlinkUpdate(HyperlinkEvent evt) {
            try {
                if(evt.getEventType() == HyperlinkEvent.EventType.ACTIVATED)
                    pane.setPage(evt.getURL());
            } catch (IOException e) {
                System.out.println(e);
            }
        }
    }

    public static void main(String[] args) {

        JEditorPane jep = new JEditorPane();
        jep.setEditable(false);
        jep.addHyperlinkListener(new LinkFollower(jep));
    }
}

```

```
try {
    URL u = new URL("http://www.gold.ac.uk");
    jep.setPage(u);
} catch (IOException e) {
    System.out.println(e);
}

JScrollPane scrollPane = new JScrollPane(jep);
JFrame f = new JFrame();
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.getContentPane().add(scrollPane);
f.setSize(512, 512);
f.show();
}
}
```

Appendix 1: Jungle Simulation

```
public abstract class Animal {

    int numberOfLegs = 4;

    public Animal(int l){
        numberOfLegs = l;
        System.out.println("In Animal.Animal()");
    }

    public int getNumberOfLegs() {
        return numberOfLegs;
    }
}

public abstract class CreepyCrawly extends Animal implements Scary {

    public CreepyCrawly(int l) {
        super(l);
        System.out.println("In CreepyCrawly.CreepyCrawly()");
    }

    public void scare(){
        System.out.println("Crawling up your leg!!");
        crawl();
    }

    public abstract void crawl();
}

public class GiantMillipede extends CreepyCrawly{

    public GiantMillipede(){
        this(1000);
        System.out.println("In GiantMillipede.GiantMillipede()");
    }
    public GiantMillipede(int l){
        super(l);
    }

    public void crawl(){
        System.out.println("In GiantMillipede.crawl()");
    }
}
```

```
public abstract class Cat extends Animal {

    public Cat(){
        super(4);
    }
    public abstract void purr();
}

public class Lion extends Cat implements Scary {

    public Lion(){
        System.out.println("In Lion.Lion()");
    }

    public void purr(){
        System.out.println("In Lion.purr()");
    }

    public void scare() {
        System.out.println("ROAR!!!");
    }

}

public interface Scary {

    public void scare();
}
```

Appendix 2: 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
```

```
public void play()
```

```
public void stop()
```