UNIVERSITY OF LONDON

GOLDSMITHS COLLEGE

B. Sc. Examination 2008

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) How does the *value* of a primitive variable differ from the *value* of a reference variable? Write a couple of assignment statements to illustrate your answer.

[5 Marks]

The value of a primitive variable is a literal value

but the value of a reference variable is the address of the object on the heap

```
int i = 3;
Box b = new Box();
```

(b) Draw a memory diagram to illustrate your answer to the above question.

[5 Marks]

Simple stack heap diagram showing i and b on the stack, a Box object on the heap.

See HFJ p 55

(c) Explain how methods are stacked by the JVM.

[5 Marks]

A stack frame is placed on the stack for every called method.

The stack frame holds the state of the method,

i.e. which line of code is executing and the values of all local variables.

The method on the top of the stack is currently running.

The top method is removed from the stack when the final left brace has been reached and execution resumes with the method immediately below.

(d) Explain, in a few lines, the concept of local and instance variable. Include in your answer an account of how the JVM allocates memory for these two types of variables.

Is there any difference in the way the JVM handles the memory allocation of primitive and reference variables?

[5 Marks]

Local variables are declared inside a method.

Instance methods are declared in a class, but outside any method.

Local variables are placed on the stack when the method is called.

Instance variables are stored with the object on the heap.

Primitive and reference variables are treated in the same way.

(e) Compare the lifetimes of local and instance variables. What is the difference between life and scope for local variables?

[5 Marks]

A local variable lives only whilst its enclosing method is on the stack.

A local variable is in scope from its point of declaration within the enclosing method, to the end of the method.

An instance variable lives as long as the object is on the heap.

A local method may enter and leave scope.

It is in scope whilst its enclosing method is on top of the stack, but goes out of scope if another method is stacked on top.

QUESTION 2

(a) Write a class `Clock` with a single instance variable `public long time` and a method `protected long now()` which returns the system time in millisecond units. The class should have a default constructor that sets the value of `time` to 0.

[5 Marks]

```
public class Clock {

    public long time;

    public Clock() {
        time = 0;
    }

    protected long now(){
        return System.currentTimeMillis();
    }
}
```

(b) Add methods `public long start()` and `public long stop()` to `Clock` so that a call to `start()` and then to `stop()` returns the duration in milliseconds of the interval between calling these two methods. The variable `time` should be reset to this duration.

[5 Marks]

```
public class Clock {

    public long time;

    public Clock() {
        time = 0;
    }

    protected long now(){
        return System.currentTimeMillis();
    }

    public void start() {
        time = now();
    }

    public long stop() {
        time = now() - time;
        return time;
    }
 }
```

(c) Write a static test method of `Clock` that times the execution of a computationally intensive block of code. You can invent any block of code to time that you wish, and you should include this code block within the test method.

[5 Marks]

```
public static void main(String[] args){
        Clock clock = new Clock();
        clock.start();
        // How long does it take to create 10 million strings?
        for (int i = 0; i < 10000000; ++i)
            new String("tick-tock");
        System.out.println(clock.stop());
    }
```

(d) What is *encapsulation*? `Clock` is not a correctly encapsulated class. Rewrite `Clock` so that it conforms to this principle.

[5 Marks]

Encapsulation means that data is hidden by marking instance variables `private`.

The data is accessed through `public` getters and setters

```
public class Clock {

    private long time;

    public long getTime(){
        return time;
    }
     // rest of class as before

}
```

(e) Extend `Clock` by writing a class `NanoClock`. You should override `now()` so that it returns the system time in nanoseconds, using the library method `public static long nanoTime()` of `System`. Include a main method for testing your new class.

[5 Marks]

```
public class NanoClock extends Clock{

    protected long now(){
        return System.nanoTime();
    }

    public static void main(String[] args){
        Clock clock = new NanoClock();
        clock.start();
        // How long does it take to create 10 milliion strings?
        for (int i = 0; i < 10000000; ++i)
            new String("tick-tock");
        System.out.println(clock.stop());
    }
}
```

QUESTION 3

(a) What are the advantages of using class inheritance in a Java project?

[5 Marks]

Inheritance means that code common to a number of classes can be put in one place.

Subclasses inherit this code from a superclass

and modifications only need to be made in one place.

A superclass defines a common protocol for a group of classes.

This keeps the design tidy and helps designers to add more classes at a later date.

(b) (i) What is an abstract method?

(ii) What is an abstract class?

(iii) Can an abstract class be instantiated?

(iv) Under what circumstances would you wish to use abstract classes?

(v) Give an example of such a circumstance.

[5 Marks]

(i) An abstract method has no body and the declaration ends in a semicolon

(ii) An abstract class has at least one abstract method

(iii) No

(iv) Use an abstract class in order to define a protocol for a group of classes, but when it doesn't make sense to instantiate the class

(v) `Animal` (abstract superclass) `Dog` (concrete subclass)

(c) Your software team has been asked to design and implement an animation of the night sky. One of your team has produced outline classes for meteorites and satellites, reproduced below:

```
public class Meteorite{

    private float xPos, yPos;
    private float xVel, yVel;
    private final float vMax = 2.0f;

    public Meteorite(float x, float y) {
        xPos = x;
        yPos = y;
        xVel = (float) Math.random() * vMax;
        yVel = (float) Math.random() * vMax;
    }

    public void move() {
        xPos += xVel;
        yPos += yVel;
    }

   // getters for xPos, yPos, xVel and yVel

    public void draw() {
        // code to draw a meteorite
    }
}

public class Satellite{

    private float xPos, yPos;
    private float xVel, yVel;
    private final float vMax = 2.0f;

    public Satellite1(float x, float y) {
        xPos = x;
        yPos = y;
        xVel = (float) Math.random() * vMax;
        yVel = (float) Math.random() * vMax;

    }

    public void move() {
        xPos += xVel;
        yPos += yVel;
    }

    // getters for xPos, yPos, xVel and yVel

    public void draw() {
        // code to draw a satellite
    }
}
```

You immediately realise that `Meteorite` ad `Satellite` have code in common, and that this code can be refactored into a higher level `abstract` class, `SkyObject`. Write a class outline for `SkyObject`.

[5 Marks]

```
public abstract class SkyObject {
    private float xPos, yPos;
    private float xVel, yVel;
    private final float vMax = 2.0f;
    public SkyObject(float x, float y){
        xPos = x;
        yPos = y;
        xVel = (float)Math.random() * vMax;
        yVel = (float)Math.random() * vMax;

    }
    public void move(){
        xPos += xVel;
        yPos += yVel;
    }
    public abstract void draw();

    // getters for xPos, yPos, xVel and yVel
}
```

(d) Write a class outline for the new version of `Satellite`, which should subclass `SkyObject`

[5 Marks]

```
public class Satellite2 extends SkyObject {

    public Satellite2(float x, float y){
        super(x, y);
    }

    public void draw(){
        // code to draw a satellite
    }
}
```

(e) Write a class `SkyTest` which can be used to test your classes. You should write a test to make sure that the satellite and meteorite objects have been correctly initialised and that the method(s) `move` function correctly.

[5 Marks]

```
public class SkyTest {
    public static void main(String[]args){

        SkyObject sat = new Satellite2(50, 75);
        SkyObject met = new Meteorite2(100, 25);
        System.out.println(sat.getX() + ", " + sat.getY());
        System.out.println(met.getX() + ", " + met.getY());
        sat.move();
        met.move();
        System.out.println(sat.getX() + ", " + sat.getY());
        System.out.println(met.getX() + ", " + met.getY());
    }
}
```

PART B

QUESTION 4

(a) Explain, with reference to call stacks, how Java threads give the impression that many processes are happening at once.

[5 Marks]

The computer can only deal with a single instruction at any one time.

This is the instruction on the top of the current call stack.

A single Java program can be threaded so that different lines of execution are placed on different call stacks.

Only one is active, the others are frozen.

The Thread scheduler switches execution from the current call stack to one of the frozen ones so quickly it gives the impression that many processes are running at once.

(b) Once a thread is running, it can move back and forth between one of three states. What are these states and under what circumstances do they occur?

[5 Marks]

Runnable: the call stack has been prepared and the thread is ready to run.

Running: the thread is active

The thread scheduler moves the thread between these two states as it tries to optimise performance of the machine

Blocked: a running thread becomes blocked when it is waiting for data

or if it has been put to sleep with a call to sleep()

or the thread is trying to call a locked method on an object

(c) Outline the steps in launching a new thread. Include some lines of code to illustrate your answer.

[5 Marks]

Make a `runnable` object `Runnable job = new MyRunnable();`

Make a Thread object (the worker) and give it a runnable (the job)

```
Thread thread = new Thread(job);
```

Start the thread

```
thread.start();
```

which places the `Runnable`'s run() method onto a new call stack.

(d) Java Servers are usually threaded. Why is this? Outline how a server might use threads.

[5 Marks]

Because, if otherwise, a blocked transaction will cause then many other connection requests to also become blocked.

The solution is to use a thread for each connection.

A server operates a pool of threads.

Incoming requests are handled by a pool thread, if there are any, otherwise it is placed in a queue.

Threads that are actively dealing with a request are removed from the pool

and replaced when the transaction is over.

(e) There are a number of dangers in thread programming. What are they?

[5 Marks]

Thread programming is difficult because threads may share memory.

This may lead to concurrency problems if two threads access the same object.

If thread 1 becomes runnable before leaving a method of a shared object, and thread 2 then accesses this object and changes an instance variable, when thread 1 resumes, the rest of its computations may be based on an out-of-date value of that variable.

Deadlock is another problem.

Here, two threads hold keys that the other threads wants. The threads will remain blocked for ever.

QUESTION 5

(a) Java uses streams to handle the transfer of data between a programme and the environment (for example, the internet or a printer).

(i) There are two categories of streams. What are they?

(ii) Give an example of a stream from each category.

(iii) Explain briefly how the two categories of stream are used to read or write data.

[5 Marks]

(i) Connection and chain

(ii) Connection: FileOutputStream; Chain: ObjectOutputStream

(iii) The chain stream deals with higher level representations of data; the connection stream acts as a helper and converts the data into bytes for the actual reading/writing operation.

(b) Write a program `SourceSaver.java` that can download the HTML source from a website into an object of type `String`. The programme should be invoked by the command

`java SourceSaver http://www.gold.ac.uk`

and must take error handling into account. (Your attention is drawn to the list of class outlines in the Appendix of this paper.)

[10 Marks]

The following code is just one solution. Award marks for any other sensible program.

```
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 SourceSaver {

    public static void main(String args[]) {

        if (args.length > 0) {
            StringBuffer buff = new StringBuffer();
            try {

                URL u = new URL(args[0]);
                Reader r = new BufferedReader(new InputStreamReader(u
                        .openStream()));
                int c = 0;
                while ((c = r.read()) != -1) {
                    buff.append((char) c);
                }

                String source = buff.toString();
            }

            catch (MalformedURLException e) {
                System.err.println(args[0] + " is not a parseable URL");
            } catch (IOException e) {
                System.err.println(e);
            }

        }
    }
}
```

(c) Write a method with the signature

`private static void writeToFile(String fileName, String text)`

which saves a `String` to file. Indicate where this method should be called from `SourceSaver` in order to save the downloaded HTML to file.

[5 Marks]

```
   .
   .
  String source = buff.toString();
  writeToFile("source.txt", source);
  .
  .
}

    private static void writeToFile(String fileName, String text) {

        try {
            FileWriter writer = new FileWriter(filename + ".html");
            writer.write(text);
            writer.close();
        } catch (IOException e) {
            System.out.print(e);
        }
    }
```

(d) Alter your code in your answer to part (b) so that all HTML tags are stripped from the source code before saving to file.

[5 Marks]

```
 int c = 0;
 boolean inTag = false;
 while ((c = r.read()) != -1) {
    if (c == '<')
       inTag = true;
```

```
  if (!inTag)
     buff.append((char) c);

 if (c == '>')
   inTag = false;


}
```

QUESTION 6

(a) Explain the purpose of `Socket`s in Java internet programming and illustrate, in a couple of lines of code, how they are used.

[5 Marks]

Sockets represent the connection between two applications

which may or may not be on two different machines (hosts).

Sockets are used for TCP communication.

A client connects to a server like this: `Socket sock = new Socket("127.0.0.1", 4200);`

Once connected, data is transferred by streams: `InputStream str = sock.getInputStream();`

(b) Explain the purpose of `ServerSocket`s in Java internet programming and briefly demonstrate how they are used in a few lines of code.

[5 Marks]

Servers use ServerSockets

to wait for client requests on a particular port.

A ServerSocket object is made `ServerSocket serverSock = new ServerSocket(4200);`

and waits for a connection `serverSocket.accept();`

returning a socket `Socket sock = serverSocket.accept();`

(c) What is object serialisation? Write a few lines for your answer, explaining what parts of an object can be serialized, and how the JVM knows that an object can be serialised.

[5 Marks]

Serialisation is a way of saving an object's state

so that the entire object graph can be read back into a Java program.

Only object variables not modified by the keyword `transient` are serialised (2 marks).

An object must implement the `Serializable` interface.

(d) Write code for an object server. This server responds to connection requests by sending a serialized object to the client. The server should not be threaded.

[10 Marks]

```
import java.io.IOException;
import java.io.ObjectOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Date;

public class SimpleObjectServer {
    public static int port = 4321;

    public static void main(String[] args) {

        try {
            ServerSocket serverSocket = new ServerSocket(port);

            boolean finished = false;
            while (!finished) {

                Socket socket = serverSocket.accept();

                ObjectOutputStream oos = new ObjectOutputStream(socket
                        .getOutputStream());

                oos.writeObject(new Date());

                oos.close();

            }
            serverSocket.close();
        } catch (IOException e) {
        }
    }
}
```

(e) [**10 Marks**]

Appendix: 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.lang.StringBuffer
public StringBuffer()
public StringBuffer(String str)
public StringBuffer append(char c)
public StringBuffer append(String str)
public StringBuffer delete(int start, int end)
public int indexOff(String str)
public insert(int offset, Srring str)
public int length()
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
pubic void play()
pubic void stop()
```