SEMESTER ONE EXAMINATIONS 2004

MODULE:	***Object-Oriented Programming for Engineers – EE553***
COURSE:	***M.Eng./Grad. Dip./Grad. Cert. in Electronic Systems*** ***M.Eng./Grad. Dip./Grad. Cert. in Telecommunications Engineering*** ***RAEC – Remote Access to Continuing Eng. Education***
YEAR:	***Postgraduate (Year 5)***
EXAMINERS:	***Dr. Derek Molloy (DCU extension 5355)***
TIME ALLOWED:	***3 hours***
INSTRUCTIONS:	Answer **FOUR** questions. All questions carry equal marks.

- Before you start, please write your name and id-number on the supplied disk.
- Please use the answer book and the supplied disks to complete your answers to this exam. For each question you attempt partly or completely electronically, please refer to it in the answer book.
- On the disk & network, please use separate directories for each question attempted, e.g. question1 etc.
- You are responsible for insuring that you have copied all the files that form your answers onto the disk and network drive. Please double check that all files are on the disk.
- All files required for the exam are on the network drive. The location will be announced at the beginning of the exam.

This booklet contains 4 pages, including the cover sheet.

DO NOT TURN OVER THIS PAGE UNTIL YOU ARE INSTRUCTED TO DO SO

Question 1.

(a) Answer the following short questions. Please keep your answers concise.

(i) Why is a *destructor virtual* in C++?
(ii) What does the *super* keyword allow in Java?
(iii) What are *namespaces* in C++?
(iv) When is the *garbage collector* called in Java?
(v) Compare the *protected* access modifier in C++ to the *protected* access modifier in Java.
(vi) What does the term *over-riding* mean?
(vii) What is an *inline method* in C++?

[14 marks]

(b) What is a *copy constructor*? What operation does the default copy constructor carry out? Using a short source-code example, demonstrate how a user-defined copy constructor can provide useful functionality.

[6 marks]

(c) Discuss *exceptions* in Java. Give an example use of exceptions when using arrays in Java.

[5 marks]

Question 2.

(a) Explain the use of *over-loaded operators* in C++. Why are they a useful feature of the language? Give a source-code example.

[8 marks]

(b) Write a *Colour class* in C++ that has the following functionality:

- The class should contain three states for red, green and blue, each with a range of 0-255.
- Add appropriate constructors.
- Is the default copy constructor appropriate?
- Write overloaded operators for +, -, =, ==, < and >.
- Write code that checks the bounds on all assignments, i.e. values cannot be <0 or >255.
- Write a sort segment of code to demonstrate your Colour class in action.

[12 marks]

(c) In C++ one class can be *a friend* of another class. Why is this useful? Show the coding syntax used to create this relationship, and discuss the advantages and disadvantages of this approach.

[5 marks]

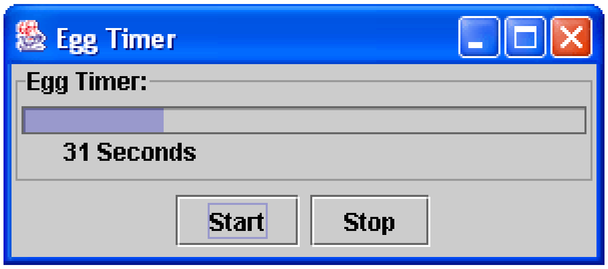
Question 3.

(a) Explain using an example why you would need to *synchronize* a segment of code when using *Java threads*? (Your answer should show a line-by-line step through of a segment of code, explaining why it would not work correctly if the segment of code was not synchronized). If synchronization is a solution to making an application thread safe, then why should we not just synchronize all our code?

[9 marks]

(b) Write a *Java egg timer* that looks like the application below.

- Once the start button is pressed the timer will count to two minutes.
- The timer can be stopped by pressing stop.
- When the timer is finished, it should pop-up an appropriate dialog box that states the time is up.



31 Seconds

[16 marks]

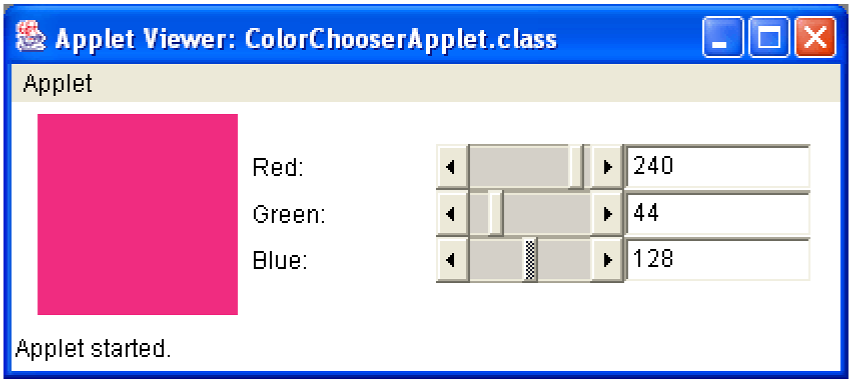
Question 4.

(a) Compare the C++ language to the Java language under the following headings only:

- *Virtual and non-virtual methods.*
- *Access specifiers.*
- *Multiple inheritance.*

[9 marks]

(b) Use the Java AWT to build the Color Chooser applet below. The applet should allow the user to choose a colour in the range (red,green,blue) of (0-255, 0-255, 0-255) using either the scrollbars or the textfields.



Red: 240

Green: 44

Blue: 128

Applet started.

[16 marks]

Question 5.

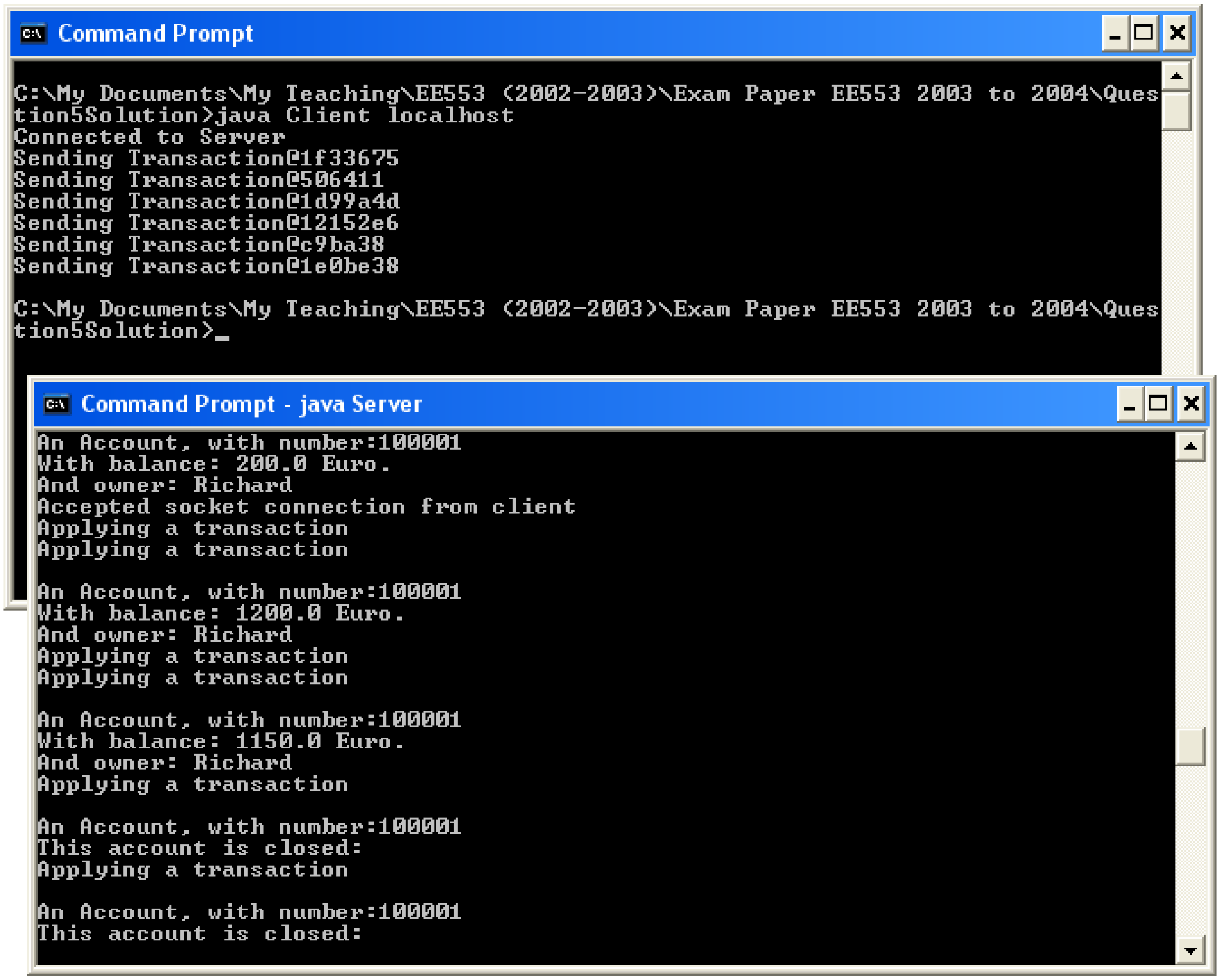
(a) Write a *Java client/server banking application*, where the client passes a Transaction object to the server, the transaction operates on an Account object on the server side, and then returns an appropriate response to the client. The basic transactions to be handled are lodgment, withdrawal, balance enquiry and close account.

You have been supplied with four classes to handle the basic aspects of this application. These are called:

- **Client.java**,
- **Server.java** (Note: this class creates three Account objects)
- **ConnectionHandler.java**
- **Account.java**

These files are in the directory **question5**. An example client/server output is shown below:

[25 marks]



```
C:\My Documents\My Teaching\EE553 (2002-2003)\Exam Paper EE553 2003 to 2004\Ques
tion5Solution>java Client localhost
Connected to Server
Sending Transaction@1f33675
Sending Transaction@506411
Sending Transaction@1d99a4d
Sending Transaction@12152e6
Sending Transaction@c9ba38
Sending Transaction@1e0be38

C:\My Documents\My Teaching\EE553 (2002-2003)\Exam Paper EE553 2003 to 2004\Ques
tion5Solution>_
```

```
An Account, with number:100001
With balance: 200.0 Euro.
And owner: Richard
Accepted socket connection from client
Applying a transaction
Applying a transaction

An Account, with number:100001
With balance: 1200.0 Euro.
And owner: Richard
Applying a transaction
Applying a transaction

An Account, with number:100001
With balance: 1150.0 Euro.
And owner: Richard
Applying a transaction

An Account, with number:100001
This account is closed:
Applying a transaction

An Account, with number:100001
This account is closed:
```