SEMESTER ONE EXAMINATIONS 2004/2005

MODULE: ***Object Oriented Programming for Engineers – EE553***

COURSE: ***M.Eng./Grad. Dip. in Electronic Systems***
M.Eng./Grad. Dip. in Telecommunications Engineering
RAEC – Remote Access to Continuing Eng. Education

YEAR: ***Postgraduate (Year 5)***

EXAMINERS: ***Dr. Derek Molloy (DCU extension 5355)***
Prof. P. Murphy (External Examiner)

TIME ALLOWED: ***3 hours***

INSTRUCTIONS: Answer **FOUR** questions.
All questions carry equal marks.

- **Before you start – put your name and id-number on the supplied disk!**
- Please use the answer books and the supplied disk to complete your answers to this exam.
- On the disk, please use separate directories for each question attempted, called question1, question2, etc.
- For each question you attempt, please reference your files on the disk related to that question in your answer book.
- **You are responsible for insuring that you have copied all the files that form your answers onto the disk – Please seek help from a technical invigil;ator if you are unsure.**

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. Keep your answers concise.

(i) Explain the difference between the terms *declaration* and *definition*?
(ii) Why are *destructors* always non-virtual in C++?
(iii) Explain how Java avoids the difficulties associated with *multiple inheritance*.
(iv) Explain the use of the *Object class* in Java.
(v) In C++ what is a *static state* and why would it be used?
(vi) What does the term *overloading* mean?
(vii) In C++ the "->" operator allows you to call a method on a pointer to an object. If this operator was not available what would you write to do the same operation?

[14 marks]

(b) Discuss the difference between arrays of objects in C++ compared to arrays of objects in Java? In particular, compare

```
SomeObject[] a = new SomeObject[5]; to:
SomeObject a[5] ;
```

[5 marks]

(c) Discuss **constructors** in C++. Can they be overloaded? Why can they not be virtual? In what order are they called when inheritance takes place? What is the copy constructor and how can it provide specific functionality?

[6 marks]

Question 2.

(a) Discuss STL Containers, Iterators and Algorithms; in particular discuss how they work together and also list different types of containers, iterators and algorithms.

[12 marks]

(b) Write a short segment of code that demonstrates a "for_each" example.

[8 marks]

(c) What are namespaces in C++? Explain the difference between the lines:

```
#include<iostream.h>   and
#include<iostream>
```

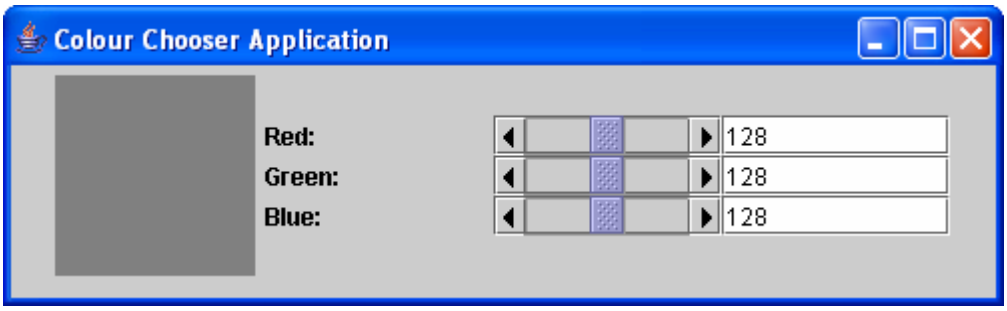
[5 marks]

Question 3.

(a) Discuss the different access specifiers available in the Java programming language.

[5 marks]

(b) Write the Java code for the application below. The square on the left-hand-side should change colour when the scrollbars are moved, or when the user types a value into the text boxes. The user should only be able to enter a valid colour value.



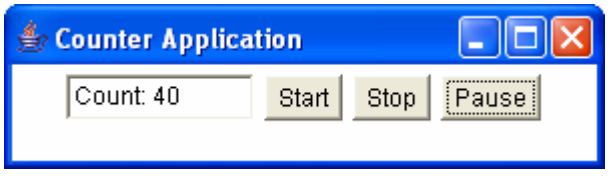
[20 marks]

Question 4.

(a) Write a short section of C++ code to demonstrate the dynamic creation of objects and to demonstrate dynamic binding working with virtual and non-virtual methods.

[10 marks]

(b) Write the Java code for the application below. It should start counting in the text field when the "start" button is pressed, should pause when the "pause" button is pressed, resume counting when the "pause" button is pressed again and stop counting when the "stop" button is pressed.



[15 marks]

Question 5.

(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?

[7 marks]

(b) Write a Java client/server application pair, where the client passes a DepositAccount object to the server and the server calculates the interest on the account using the current interest rate available on the server. The account object is then passed back to the client, where the client displays the details including the updated balance. The server should also display the amount of money added through the interest calculation.

You have been supplied with three sets of code to handle the basic aspects of this application. These are called:

- **Client.java**,
- **Server.java** and
- **ConnectionHandler.java**

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

[18 marks]