DUBLIN CITY UNIVERSITY

SEMESTER ONE EXAMINATIONS 2009

MODULE: ***(Title & Code)***	Object-oriented Programming for Engineers EE553
COURSE:	M.Eng./Grad. Dip./Grad. Cert. in Electronic Systems M.Eng./Grad. Dip./Grad. Cert. in Telecoms. Eng. IPME – Individual Postgraduate Modules – Electronics.
YEAR:	Postgraduate(C)
EXAMINERS:	Prof. P. Rees (External Examiner) Dr. Derek Molloy (DCU - Ext.5355)
TIME ALLOWED:	3 Hours
INSTRUCTIONS:	Please answer FOUR questions. All questions carry equal marks

Requirements for this paper
Please tick (X) as appropriate

□ ***Log Table***
□ ***Graph Paper***
□ ***Attached Answer Sheet***
□ ***Statistical Tables***
☒ ***USB Key (supplied by lecturer)***

THE USE OF PROGRAMMABLE OR TEXT STORING CALCULATORS IS EXPRESSLY FORBIDDEN

• Please use the answer book and the supplied USB key to complete your answers to this exam. For each question you attempt partly or completely electronically, please refer to it in the paper answer book. Please write your ID number on the USB key tag.
• On the USB key & network, please use your ID number as the root directory and place separate directories for each question attempted in this directory, e.g. Q1 etc.
• You are responsible for ensuring that you have copied all the files that form your answers onto the USB key and network drive. Please double check.
• Any additional files required for the exam are on the network drive. The location will be announced at the beginning of the examination.

Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones.

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

Question 1

1(a) Answer the following questions (keep your answers concise): [14 marks]

(i) Describe the use of the **this** keyword in Java.
(ii) How is the **new** keyword different in C++ to Java?
(iii) Describe the difference between the role of a **compiler** and an **interpreter**?
(iv) Describe how a **union** structure works in C++?
(v) What is the main difference between **abstract classes** in C++ and abstract classes in Java?
(vi) Describe the use of the conditional operator **'?'** in C++.
(vii) Explain the use of the following segment of code:

```
template <class T>
T add(T a, T b)
{
  return a+b;
}
```

What will happen if + is undefined for a particular type used?

1(b) Examine the following section of code: [6 marks]

```
class A
{
   private:
     int x;
     friend class B;
 };

class B
 {
   x(A &a)
   {
     a.x++;  // Point 1
   }
 };

 class C: public B
 {
   y(A &a)
   {
     a.x++;  // Point 2
   }
 };
```

Why does the code work correctly at 'Point 1' and why does it fail at 'Point 2'? Why are 'friends' used in C++?

1(c) Discuss the use of **Non-Virtual** methods in C++. Why are they used? In C++, why is non-virtual default? Does Java have non-virtual methods, or an alternative to them? [5 marks]

[Total marks: 25]

Question 2

2(a) Examine the following class definitions: [16 marks]

```
#include <iostream>
#include <string>
using namespace std;

class Vehicle
{
   string color;
   string brandName;
  public:
   Vehicle(string, string);
   virtual void display() = 0;
};

class Car: public Vehicle
{
   int numberSeats, numberDoors, numberWheels;
  public:
   Car(string, string, int, int, int);
   Car(Vehicle, int, int, int);
   virtual void display();
};
```

Write an implementation for each of the methods listed in the class definition. Write a main() function that would test the methods and all constructors. Write a class for a Motorbike class that fits into the example above.

2(b) Examine the following segment of code that has several errors: [9 marks]

```
#include<iostream>

class Calculator
{
        virtual float total;

  public:
        virtual Calculator(float);
        virtual float add(float);
        virtual float multiply(float);
        virtual float subtract(float);
        virtual float divide(float);
        virtual float getTotal();
        virtual void setTotal();
};

Calculator::Calculator(float a): total(a) {}
float Calculator::add(float a)  {return total+=a;}
float Calculator::multiply(float a)  {return total*=a;}
float Calculator::subtract(float a)  {return total-=a;}
float Calculator::divide(float a)  {return total/=a;}
float Calculator::getTotal()  {return total;}
void  Calculator::setTotal(float a)  {total = a;}

void Calculator::main()
```

[PTO]

```
26      {
27              Calculator c(100), d();
28              c.add(50.0f);
29              c.divide(10.0f, 2.0f);
30              d.add(20.0f);
31              cout << "The value of c is: " << c.getTotal() << endl;
32              cout << "The value of d is: " << d.total << endl;
33              return 0;
34      }
```

Locate the errors (there are approx. 9) and describe why you believe there is an error at that location. Use the line numbers to help you to explain your answers.

[Total marks: 25]

Question 3

3(a) Write the implementation for the following class definitions and write a main() function to test them. [10 marks]

```
class Person {
        string name, id;
  public:
        Person(string, string);
        virtual void display();
        virtual string getRole() = 0;
};

class Student: public Person  {
        string programme;
        int year;
  public:
        Student(string, string, string, int);
        virtual void display();
        virtual string getRole();
};

class Lecturer: public Person  {
        string office;
        int phoneNum;
  public:
        Lecturer(string, string, string, int);
        virtual void display();
        virtual string getRole();
};
```

3(b) Write a template storage container that is capable of storing a specified number of generic objects. It should have the capability to return the number of objects in the store, to return an indexed object and a simple mechanism for adding an object to the end of the store. [7 marks]

3(c) Use the storage container from (b) to create a Person store that is capable of storing both Student and Lecturer objects. Write [3 marks]

[PTO]

code to test this store.

3(d) Use the STL vector class to perform the same tasks as your storage container and repeat part (c) using the STL vector. [5 marks]

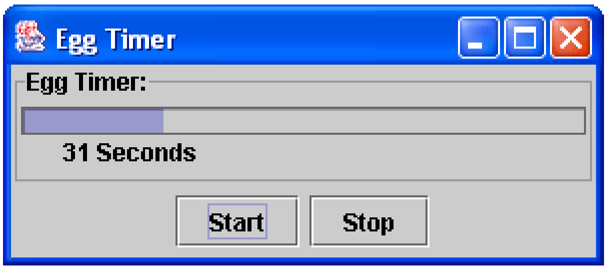
[Total marks: 25]

Question 4

4(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]

4(b) Write a Java egg timer that looks like the application below. [16 marks]

- 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.



[Total marks: 25]

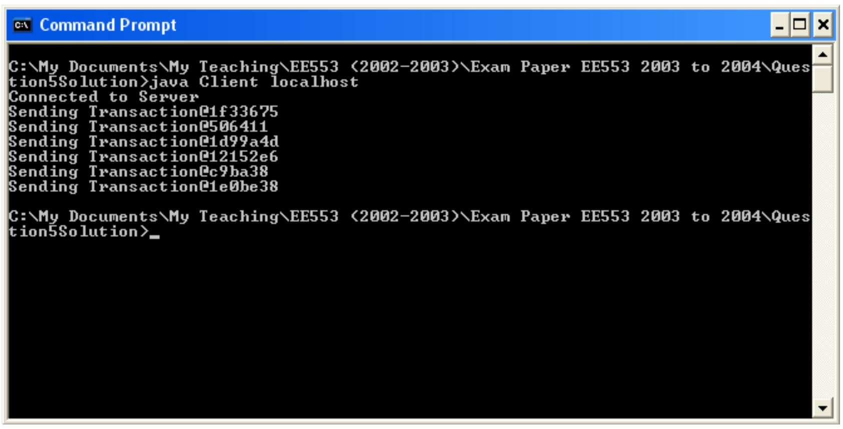
Question 5

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 lodgement, withdrawal, balance enquiry and close account. You have been supplied with four classes to handle the basic aspects of this application. These are called: [25 marks]

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

[PTO]

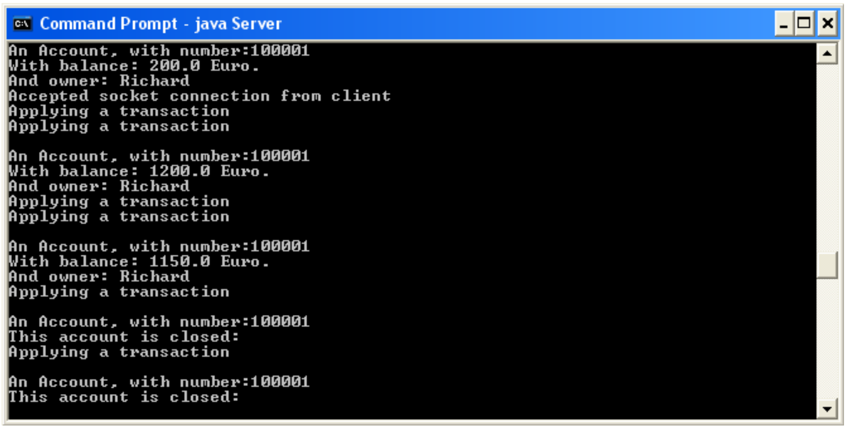
Files are in the directory **question5**. An example client/server output is shown below:



```
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>_
```

Figure 5.1 The Client



```
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:
```

Figure 5.2 The Server

[Total marks: 25]