

DUBLIN CITY UNIVERSITY

SEMESTER ONE EXAMINATIONS 2009/2010

MODULE:	EE553 Object-oriented Programming for Engineers
COURSE:	 MTCC – M.Eng. in Telecommunications Eng. MEN – M.Eng. in Electronic Systems IPME - Individual Postgrad. Modules-Electronics MEQ - Masters Engineering Qualifier Course GCES – Grad Cert. in Electronic Systems GCTC – Grad Cert. in Telecommunications Eng. GDE – Graduate Dip. in Electronic Systems GTC – Graduate Dip. in Telecommunications Eng.
YEAR:	C
EXAMINERS:	Prof. Peter Ashburn Dr. Derek Molloy, Ext no. 5355
TIME ALLOWED:	3 Hours
INSTRUCTIONS:	Answer Four Questions

Please do not turn over this page until you are instructed to do so

The use of programmable or text storing calculators is expressly forbidden.
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

This is a computer based examination:

- Please use the answer book and the supplied USB key to complete your answers. For each question you attempt partly or completely electronically, please refer to it in the paper answer book.

- On the USB key & network, please use your ID number as the root directory and use separate subdirectories for each question attempted 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.

EE553 Object-oriented Programming for Engineers Semester One Examinations 2009/2010 PAGE 1 OF 6

QUESTION 1

[TOTAL MARKS: 25]

1(a)

[16 Marks]

Answer the following questions (keep your answers concise):

- (i) In Java, why is the *super()* call always the first line of the child's constructor? How does this compare to C++?
- (ii) Why are all methods in the Java Math class static?
- (iii) Why do pointers require a *dereference type* in C++?
- (iv) What is the main difference between a C++ class and a C++ struct?
- (v) What is the main difference between abstract classes in C++ and abstract classes in Java?
- (vi) Describe the use of the *break* and *continue* keywords in C++.
- (vii) Why can abstract classes not have static abstract methods?
- (viii) Briefly explain the difference between *modal* and *modeless* dialog boxes in Java.

1(b)

[4 Marks]

The following lists two segments of code (C++ and Java):

C++ code:

Account a(); CurrentAccount b(500,5000); a = b;

Java Code:

Account a; CurrentAccount b = new CurrentAccount(500,5000); a = b;

where Account is the parent class of CurrentAccount. How does the result of the a=b operation differ in this case between C++ and Java?

1(c)

[5 Marks]

Discuss the use of non-virtual methods in C++. In C++, why is non-virtual the default? Does Java have non-virtual methods, or an alternative to them?

[End of Question 1]

EE553 Object-oriented Programming for Engineers Semester One Examinations 2009/2010

PAGE 3 OF 6

QUESTION 2

[4 Marks]

Describe the use of the *void* pointer in C++ through the use of a source code example. How can we modify a value that is pointed to by such a pointer?

[8 Marks]

[TOTAL MARKS: 25]

Write method implementations for the Person class definition below:

#include<iostream> #include<strina> using namespace std; class Person { private: string name; friend void clearName(Person &); public: Person(string); Person(const Person &); virtual bool operator == (Person); virtual void display(); };

2(c)

[8 Marks]

Write a child class "Student" for the class described in 2(b) that should differ slightly from the Person class. The child class should also contain an overriding display() method. Write a main() function that tests all of the methods described in the Person class and your Student class.

2(d)

[5 Marks]

What is dynamic binding? Add some code to the main() function in 2(c) to demonstrate the concept of dynamic binding using the Person and Student classes.

[End of Question 2]

2(a)

2(b)

QUESTION 3

[TOTAL MARKS: 25]

3(a)

[7 Marks]

[7 Marks]

Describe the following C++ STL categories: Containers, Iterators and Algorithms. Concisely describe four examples for each category.

3(b)

Write some C++ code that stores values of any standard type (e.g. int, float, double etc.) in a STL vector, uses a STL algorithm to sort the data and then displays the values using a STL for_each algorithm. Note: the output function should work for any standard type.

[11 Marks]

Write a section of Java code that demonstrates the following features of the Java Vector class (java.util.Vector) in the following order:

- Add objects of any class to the Vector.
- -Demonstrate the use of the capacity() method and show how the capacity changes in a vector when the current capacity is exceeded.
- Use the contains() method.
- Remove the last element from the vector.
- Use the setElementAt() method.
- Use the *lastIndexOf()* method.
- Display the remaining contents of the Vector.

Attach the output of your code to your answer.

[End of Question 3]

QUESTION 4

[TOTAL MARKS: 25]

[5 Marks]

[10 Marks]

Discuss the differences and relationships between: a Window; a Frame; a Dialog Box; and a Panel in Java. Java Graphical User Interface (GUI) applications are event driven - what does this mean?

4(b)

4(a)

Write the Java code to create the Java Swing Application that is displayed in Figure 4.1. The application should allow a user to enter text in a text field that will be displayed in the text area when the send button is pressed. If the clear button is pressed the text that was being entered in the text field is cleared. This must be written as a Swing application and the application should terminate when the X button is pressed.

[PTO]

EE553 Object-oriented Programming for Engineers Semester One Examinations 2009/2010

PAGE 4 OF 6

3(c)



Figure 4.1

4(c)

[10 Marks]

Modify the Java Swing application written in part (a) to allow any number of windows to be opened at the same time. When a message is typed in any window's text field and send is pressed, the message will be displayed in the text area of all windows. The number of windows could be passed from the command line interface. See Figure 4.2 below. Note: This is <u>not a client/server</u> <u>application</u> as the windows will all be on the same JVM.

🛃 Q4 Window 0	
Window 0 sends out a message Window 1 sends out a message Window 2 sends out a message	
Window 0 about to send out another message Send Clear	
🛃 Q4 Window 1	
Window 0 sends out a message Window 1 sends out a message Window 2 sends out a message	
Send Clear	
🙆 Q4 Window 2	
Window 0 sends out a message Window 1 sends out a message Window 2 sends out a message	
Window 2 is also about to send out another message	
Send Clear	

Figure 4.2

[End of Question 4]

QUESTION 5

[TOTAL MARKS: 25]

5(a)

[25 Marks]

Write a <u>client/server</u> application that allows a GUI client to send a message to the server that is converted to lower case and sent back to the client. There should be a suitable graphical user interface. See Figure 5.1 for an example client interface and Figure 5.2 for an example running server. The server should be made <u>fully threaded</u>.

Note: You have been provided with 3 files to help you with your solution: Client.java, Server.java and ConnectionHandler.java on the examination network drive. Please build your solution using these files.

실 Office-PC/192.168.1.7	- 0 <mark>- X</mark>	
Question 5 Converts Strings From UPPER CASE to LOWER CASE question 5 converts strings from upper case to lower case Using The Server To Do The Conversion - YOU CAN SEE THIS ON TH	HE COMMAN	
using the server to do the conversion - you can see this on the comm	and shell	-
using the server to do the conversion - you can see this on the comm	and shell	•
Using the server to do the conversion - you can see this on the comment I lippe my UPPERCASE Message Here and Press Send to sent it to th	e Server	•

Figure 5.1 Client GUI, where the text is entered and server results are displayed.



[End of Question 5]

[END OF EXAM]

EE553 Object-oriented Programming for Engineers Semester One Examinations 2009/2010 PAGE 6 OF 6