CARDIFF UNIVERSITY
EXAMINATION PAPER

Academic Year:	2013/2014
Examination Period:	Spring
Examination Paper Number:	CMT205
Examination Paper Title:	Object Oriented Development with Java
Duration:	2 hours

Do not turn this page over until instructed to do so by the Senior Invigilator.

Structure of Examination Paper:

There are 4 pages.
There are 4 questions in total.

The maximum mark for the examination paper is 60 and the mark obtainable for a question or part of a question is shown in brackets alongside the question.

Students to be provided with:

The following items of stationery are to be provided:
ONE answer book.

Instructions to Students:

Answer THREE questions.
Students are permitted to introduce to the examination any textbook, any printed / handwritten notes, and other similar materials. Use of annotations, highlighting and bookmarks is permitted.
The use of calculators is permitted in this examination.
The use of translation dictionaries between English or Welsh and a foreign language bearing an appropriate departmental stamp is permitted in this examination.

Q1. (a) What are the *values* of the variables `res1`, `res2`, `res3`, `res4` and `res5` after the following Java statements have been executed? [5]

```
int inum = 23;
double dnum = 5.4;
double res1 = (double) inum / 5 + inum / 4;
double res2 = (double) (inum / 3 + dnum * (1/3) );
int res3 = (int)dnum << 2;
String res4 = inum + "+" + dnum;
StringBuffer buffer = new StringBuffer(res4);
buffer.insert(1, "-");
buffer.append("=");
String res5 = buffer.toString();
```

(b) Write Java statements for the following tasks:

i. Fill a square of size 10×10 pixels in *blue* with the top left corner at coordinates $(100, 120)$, given a Graphics object `g`. [2]

ii. Define and initialise an array `num` with 100 random double numbers between 0.0 (inclusive) and 1.0 (exclusive). [2]

iii. Given a class named `TaskThread` which implements the `Runnable` interface and *two* instances of this class `t1` and `t2`, create and start *two* threads each running one of these instances simultaneously, and wait until both threads finish. [3]

(c) Write Java statements to get an integer from keyboard input and store the result in an `int` variable `ival`. If the user input is not an integer, allow the user to try until a proper integer is entered. You **do not need** to print prompt messages. You may assume that `IOException` will be handled elsewhere. [5]

(d) To allow some arbitrary operation to be applied to all the elements in a collection, what design pattern would be suitable? Briefly explain the basic idea of how to implement this design pattern in Java (no more than three sentences). [3]

Q2. (a) Describe how you may create a Combo box that gives a selection of countries. E.g Wales, Scotland, England, Ireland, France and Germany. How may you add and delete countries from this list? [7]

(b) An application is required to *filter out* all references to web sites that end with the extension `.inappropriate.com`. Having detected a suitably inappropriate web site it should *inform* the user with a suitable prompt.

Show, with the aid of suitable diagrams and code fragments only, how you would achieve this with the appropriate Java Swing Components. Your solution should only focus on the mechanisms that achieve the filtering and informing operations — You **do not need** to develop a complete GUI. [13]

Q3. (a) Suppose we wish to store student records in a structured binary file. Each record contains the following fields (field name and type):

`Number:` `int` (the student number, starting from 1)

`Name:` `String` (the full name, with up to 30 characters)

`Marks:` `double` (the student marks)

Assume that the student number is more or less allocated sequentially. There are 100 students currently enrolled and the maximum student number is 120. The student name on average takes 15 bytes when represented in UTF-8 (including the bytes corresponding to the string length).

i. How many bytes will be needed to store all the records using a sequential access file and a random access file, respectively? [4]

ii. If records need to be updated frequently, which one will be more suitable: a sequential access file or a random access file? Why? [2]

(b) Complete the following Java program `Count` that reads in a text file and counts how many alphabetic letters and words appear in the text file. For simplicity, a word is defined as a maximum consecutive sequence of alphabetic letters (separated by any non-alphabetic characters). For example, `That's good.` is considered to have three words (`That`, `s` and `good`). The program takes a single command line argument that specifies the filename of the text file to be processed. You should handle exceptions properly. You can assume that relevant Java standard library classes have already been imported. [14]

The following code is provided and only the missing code needs to be completed:

```
import java.io.*;
public class Count
{
    public static void main(String[] args)
    {
        int countLetters = 0;    // count of total letters
        int countWords = 0;      // count of total words

        // check if one command line argument is provided
        if (args.length != 1)
        {
            System.out.println("One command line argument needed!");
            System.exit(-1);
        }

        // TODO: complete the code here
        // ...

        // Print the numbers of letters and words in the text file
        System.out.println("Total letters = " + countLetters);
        System.out.println("Total words = " + countWords);
    }
}
```

Q4. Complete the following Java program `EncServer` that implements a UDP server which accepts UDP packets at port 6000. Each packet received is a string containing a text message to be encrypted. For each received packet, the server sends back a packet containing a string corresponding to the encrypted message using the following rule: replace each lower case letter with the next letter cyclically, i.e. 'a' becomes 'b', 'b' becomes 'c', ..., and 'z' becomes 'a'. Punctuation, any unexpected uppercase letters, and any other non-alphabetic characters should be left unchanged. You can assume that the incoming packets are in the right format but exceptions need to be handled properly. You can assume that relevant Java standard library classes have already been imported. [20]

The following code is provided and only the missing code needs to be completed:

```
import java.io.*;
import java.net.*;
public class EncServer
{
    public static void main(String[] args)
    {
        // TODO: complete the code here
        // ...
    }
}
```