CARDIFF UNIVERSITY
EXAMINATION PAPER

Academic Year:	2014/2015
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 5 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.
Important note: if you answer more than the number of questions instructed, then answers will be marked in the order they appear only until the above instruction is met. Extra answers will be ignored. Clearly cancel any answers not intended for marking. Write clearly on the front of the answer book the numbers of the answers to be marked.

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 `result1`, `result2`, `result3`, `result4` and `result5` after the following Java statements have been executed? [5]

```
int inum = 19;
double dnum = 2.6;
double result1 = (double) inum / 5 + dnum;
double result2 = (double)(inum % 3 + (1/2) * dnum);
String result3 = inum + "+" + dnum;
StringBuffer buffer = new StringBuffer(result3);
buffer.insert(2, "abc");
String result4 = buffer.toString();
buffer.replace(1, 4, "to");
String result5 = buffer.toString();
```

(b) Write Java code fragments for the following tasks:

i. Draw a filled disc in **blue** with centre at $(60, 80)$ and a radius of 40, given a Graphics object `g`. [2]

ii. Define a hash map `StuMarks` that maps student names to their marks, assuming each student name is represented as a `String` and each mark as an `int`. Add an entry to the map to indicate `Bill` has got 60 marks. [2]

iii. Given a `double` array `num`, print the minimum number in the array. [3]

(c) Assuming a remote server implements the following interface

```
import java.rmi.*;
public interface DistCalc extends Remote
{
    public double CityDist(String city1, String city2)
                    throws RemoteException;
}
```

which calculates the distance in miles between two cities (with `city1` and `city2` being the name of two cities). The RMI service is registered with a URL of `rmi://rmi.cs.cf.ac.uk/dist`

Complete the following code to print the distance between Cardiff and London using RMI. [4]

```
import java.rmi.*;
public class DistClient  {
   // main method
   public static void main( String[ ] args )  {
       // TODO: complete your code here
   }
}
```

Question continues on next page

(d) Assume you are developing a Java program for a water supply company. The program involves a class named `Account` that represents a generic user account, a class named `FlatRateAccount` that represents a user paying a flat rate (independent of their usage) and a class named `MeteredAccount` that represents a user whose charge is based on the amount of water they use.

i. When you define these classes, what are the relationships between them (if any)? [1]

ii. Define a method `CalcBill` to work out the bill for each user, for all the user accounts. Briefly explain how you defined this method and how you would implement it in these classes. [3]

Q2. (a) Describe, using suitable fragments of Java code, how you would create a `JTable` with the following elements:

First Name	Last Name	Student Number	Year of Study	Date of Birth

Clearly state which are the data types of each element.

Illustrate how you populate the table by adding *two* rows to your table. [5]

(b) For the table given above, show (with suitable fragments of Java code) how you implement a method to sort a selected column of the table. You should be able to order the data numerically or lexically according to the data type.

Your solution should only detail how you control a `JTable` to get user selection and display of data once sorted. You may assume that a data sorting method exists for any data type in your `JTable`'s columns: You do not need to create a sorting algorithm. [15]

Q3. (a) Suppose we wish to store book records for a library in a structured binary file. A record includes the following fields (name and type):

- `bookID`: `int` (the ID of the book)
- `title`: `String` (the title of the book, up to 60 characters)
- `publisher`: `String` (the publisher of the book, up to 30 characters)
- `price`: `double` (the price of the book)

We further assume that `bookID` is allocated sequentially, starting from 1. However some books may later be withdrawn from the library. At the moment, there are 2000 books in the library and the maximum `bookID` is 3000. The fields `title` and `publisher` when represented in UTF-8 take on average 30 and 10 bytes respectively.

i. If a sequential access file is used, work out the space (number of bytes) each record takes on average, and the total number of bytes needed. [3]

ii. If a random access file is used, work out the space (number of bytes) each record takes, and the total number of bytes needed. [3]

iii. Assuming we now need to backup all the book records to some cloud storage (for later recovery if needed), which type is more suitable, sequential or random access file? Justify your answer. [2]

(b) Write a Java program that takes a text file as input and produces a new text file containing the content from the input text file but with characters in each line in a reversed order. For example, if the input text file contains

```
abc
123
```

The output text file should contain

```
cba
321
```

Your program takes two command line arguments, giving the input and output file names respectively. The text file may contain general Unicode characters. You can assume all the necessary classes from the Java standard library are imported. The following skeleton code is provided and only the code to complete the program needs to be provided. [12]

```
import java.io.*;
public class TextReverse
{
    public static void main(String[] args)
    {
        if (args.length != 2)
        {
            System.err.println("Two arguments expected.");
            System.exit(-1);
        }

        // TODO: Complete the code
    }
}
```

Q4. (a) Write a Java program `TransServer` that receives information of transactions, each containing the unit price and the number of units purchased, and sends back the running total (the total amount based on the transactions received). Your program should listen to incoming **TCP** connections at the port 8000. For simplicity only one connection needs to be dealt with at a time. Every time a new connection is made, the running total is set to 0. While the connection keeps alive, the client sends a line at a time, containing two `double` numbers, separated by blankspace(s). The first number gives the unit price and the second number gives the number of units purchased. The server should send back a line containing a single number, corresponding to the running total based on the transactions received so far.

You may assume that the content received from the client is always in the correct format but communication exceptions need to be handled properly. You can assume all the necessary classes from the Java standard library are imported.

Example:

Received:

```
3.0 2
5.2 1.5
```

Send back:

```
6.0
13.8
```

Here, $6.0 = 3.0 \times 2$, and $13.8 = 3.0 \times 2 + 5.2 \times 1.5$.

The following skeleton program is provided and only the code to complete the program needs to be provided. [16]

```
import java.net.*;
import java.io.*;
import java.util.*;
public class TransServer
{
    public static void main()
    {
       // Create a Server Socket object
        ServerSocket sSock = null;
        try
        {
            sSock = new ServerSocket( 8000 );
        }
        catch( IOException e )
        {
            System.err.println(e);
            System.exit(1);
        }

        // TODO: Complete the program
    }
}
```

(b) What changes will be needed so that the server can handle multiple connections at the same time? Briefly describe your approach. You don't need to write the Java program but you need to cover all the major implementation ideas. [4]