Autumn Examinations 2013

Exam Code(s)	3IF1, 3BP1
Exam(s)	Third Year Information Technology Third Year Electronic and Computer Engineering
Module Code(s)	CT326
Module(s)	Programming III
Paper No.	1
External Examiner(s)	Prof. M. O'Boyle
Internal Examiner(s)	Prof. G. Lyons Dr. M. Madden *Dr. D. Chambers
Instructions:	Answer any 4 questions. All questions carry equal marks.
Duration	3 hrs
No. of Pages	5
Department(s)	Information Technology
Requirements	None

1. Develop a simple Java based payroll system that can calculate the weekly pay due for different categories of employees. The system should be implemented using the following design guidelines:

a: Implement an *abstract* base class called Employee that is used to hold and access basic information about an employee e.g. name, address, etc. This class should also define an *abstract* method called earnings() that returns the weekly pay for each type of employee. The class should include a suitable constructor and accessor methods to retrieve information about the employee.
5 MARKS

b: Implement a class called Manager, derived from Employee. A manager is paid a fixed weekly salary. The class should include a suitable constructor and should also implement the earnings() method. 5 MARKS

c: Implement a class called HourlyWorker, derived from Employee. An hourly worker is paid a fixed wager per hour, so in any given week they will be paind for the number of hours worked in the past week. The class should include a constructor and implement the earnings() method. 5 MARKS

d: Implement a class called CommissionWorker, derived from Employee. A commission worker is paid a base salary per week and an additional bonus based on the number of items sold during the past week. The class should include a constructor and earnings() method. 5 MARKS

e: Write a short driver program that creates an object for each of the employee sub-classes, it then calls the earnings() method for each object and displays the results. 5 MARKS

2.a: Describe the general structure and purpose of the IO Streams classes provided in the Java programming environment. What Java Class is used to support random file access? 4 MARKS

b: Write a Java application that inputs a date as a string in the form 23-03-2013 The program should use an object of class *StringTokenizer* to extract the various components of the date string as tokens. The program should then convert the day, month and year to int values and display them. 6 MARKS

c: Write a simple Employee class that includes an id number, a name, and salary details and a suitable constructor method. Then write a Java program that uses an ArrayList to store a collection of Employee objects. Also, write the code for a Comparator class i.e. a class that implements the Comparator interface, that can be used to compare two Employee objects based on their id number. Finally, use the version of the Collections.sort() method that allows you to pass your own Comparator object to sort the list of Employee objects. 15 MARKS

3.a: What types of Sockets are supported in the Java networking package? Which type of Socket would you recommend for a VOIP type application and a File Transfer type application? 5 MARKS

b: Write a Java application that uses Stream type sockets to exchange Java Objects using object serialisation. The client side should connect to the server and send it a String object. The server should print out the String and respond to the client with a text based response encapsulated in another String Object. The client should receive the String Object from the server and print out this response. 10 MARKS

c: Write another Java application with the same functionality as outlined above, in part b of this question, but this time using Datagram type sockets. Hint: you can use ByteArrayOutputStream and ByteArrayInputStream to populate and read the array associated with the DatagramPacket object. 10 MARKS

4.a: Discuss briefly the differences between a process and a thread. What is the best way to stop executing threads be stopped (assuming they still haven't finished their work)? 5 MARKS

b: Write a JAVA animation applet that uses a thread to continuously scroll a text message across the screen from right to left. The message itself and the rate at which the text scrolls can be passed to the applet as HTML based parameters. 10 MARKS

c: Outline the design and code implementation of the Java class for an object that will be used as a buffer to hold an integer value. The value may be updated randomly by one or more Producer threads, provided that it has already been consumed by one of a number of Consumer threads. Each value produced must be consumed at exactly once and there may be multiple producer and consumer threads executing (and attempting to access the buffer) concurrently. 10 MARKS

5. Assume that a Sports Club at the University wishes to store details about its members. Design and implement a Java application to support this requirement. The application should be able to print out and manage information about the members of the club. The following guidelines should be used to construct the application:

a: A Java class, called Member, should be defined to store and manage student details. The class should include methods for updating member details and querying their registration status i.e. are they fully paid up members of the club. Each member of the club should also have a unique membership id number, this number is automatically assigned when the member object is created. 10 MARKS

b: Define another Java class, called SportsClub, that will be used to manage club membership and access details about individual members. Member objects added to the SportsClub should be stored using a suitable collection object. SportsClub should include methods for adding new members, removing members, getting a list of current members and accessing information about an individual member (based on their name or id number). 10 MARKS

c: Write a short driver program, in a class called ClubManager, that creates an instance of SportsClub and uses its methods to add, lookup and remove club members. 5 MARKS

6. Using Java Remote Method Invocation, write the Java code for a remote compute server that could be used to remotely execute arbitrary Task objects. The server allows clients to submit Task objects, that is objects that implement the Task interface, for remote execution on the server and are then returned the result as a Java object. The following Java interfaces / classes should be provided:

   - *Compute* - this remote interface should provide a method to upload Task objects to the server and to then run the task and return the result back to the client when execution is complete. 4 MARKS

   - *Task* - this interface should define an arbitrary task object that may be passed as a parameter to the compute server. 4 MARKS

   - *MathTask* – this class provides an implementation of the Task interface and is used to perform some calculation that returns an Integer object. The calculation itself can be just some simple arithmetic e.g. add two numbers. 6 MARKS

   - *ComputeServer* - this class should provide an implementation of the Compute interface as well as the code required to initialise the server and make the remote object locatable for clients in the RMI registry. The server runtime should be protected so that objects uploaded to the server can not cause any harm. 6 MARKS

   - *ComputeClient* – this should provide a simple client program that creates a MathTask object and submits it to the server for remote execution and then displays the result. 5 MARKS

   The design of the system should make it possible for new Task classes to be easily added to the system in the future, making the system very flexible. The design should use Java RMI and Object Serialisation to submit Task objects and to return the result back to the client.