Semester 1 Examinations 2017 / 2018

Exam Code(s)	3BCT, 3BP, 3BLE
Exam(s)	Third Year Computer Science & Information Technology Third Year Electronic and Computer Engineering Third Year Electrical and Electronic Engineering
Module Code(s)	CT326
Module(s)	Programming III
Paper No.	1
External Examiner(s)	Dr. J. Howe
Internal Examiner(s)	Dr. M. Schukat *Dr. D. Chambers
Instructions:	Answer any 4 questions. All questions carry equal marks.
Duration	2 hrs
No. of Pages	4
Department(s)	Information Technology
Requirements	None

1. The following Java code provides the outline of a simple bank account class:

```
import java.io.*;
public class Account implements Comparable<Account>,
Serializable {
     protected int accnum;
     protected HolderDetails holder;
     protected List<Transaction>  transactions;
     protected float balance;

     // Add a suitable constructor here

     // Add methods to make deposits / withdrawals

     // Method to print out account transaction summary

     // Add suitable attribute accessor methods

     // Add method to implement the Comparable interface
}
```

a: Complete the implementation of the Account class, providing a suitable constructor, attribute accessor methods, methods for making a deposit or withdrawal, a method to print out a transaction summary related to a range of dates and an implementation method for the Comparable interface.
7 MARKS

b: Provide implementations for the HolderDetails class and the Transaction class. The HolderDetails class is used to store personal details about the account holder. The Transaction class contains details about past transactions including the type of transaction, the amount and the Date. 7 MARKS

c: Define and implement a new class, called CurrentAccount, derived from Account, that allows withdrawals to proceed up to some overdraft limit. Note that the base Account class shown has no overdraft facility.
5 MARKS

d: The attributes of class Account are defined as *protected*. What is the implication of this definition? 3 MARKS

e: What does the statement *implements Serializable* mean? What are the implications of this statement? 3 MARKS

2.a: Implement an enum in Java which enumerates the days of the week. Assuming the week starts on Sunday, include in the enum the relevant position of the day in the week i.e. Sun = 1, Mon = 2, etc and whether the day is a normally a working day i.e. Sun = false, Mon = true, etc. Provide a suitable toString() method to print information about the enumerated types. 12 MARKS

b: 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. 13 MARKS

3.a: Write a network server program in Java where the server waits for incoming client connections using stream type sockets. Once a client connects it sends a String object to the server with a simple query – the server then responds with a text based response. The connection is then terminated. The server should use a separate thread of execution for each new client connection and all interaction between the server and the client should be done within this thread. The answer only needs to include source code for the server side application. 12 MARKS

b: Write another Java application with the same functionality as outlined above, in part a 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. This application does not need to implement a reliable data transfer protocol. The answer only needs to include source code for the server side application. 13 MARKS

4.a: Discuss briefly the differences between a process and a thread. What is the best way to stop executing threads, assuming they still have not 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 give the full source code for a Java class that will be used as a thread safe 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 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 member 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. The Member class should implement the Comparable interface and use the membership id number to define the natural order for these objects.
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. Member objects should be sorted by their id number as they are added to the collection. SportsClub should include methods for adding new members, removing members and returning a list of current members. 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. The program should also be able to save the full application state on exit, using Object Serialization, and load up the saved state at startup.
5 MARKS