NUI Galway
OÉ Gaillimh

Semester 1 Examinations 2018 / 2019

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)	Prof. M. Madden *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 {
        protected HolderDetails holder;
        protected List<Transaction> transactions;
        protected float balance;

        // Add a suitable constructor here

        // Add methods to make deposits / withdrawals

        // Add a method to print out summary of account transactions

        // Add suitable accessor methods for retrieving / updating attributes
}
```

a: Complete the implementation of the Account class, providing a suitable constructor, attribute accessor methods, methods for making deposits / withdrawals and a method to print out a transaction summary for a given time interval. Include appropriate user defined Exceptions as required.
8 MARKS

b: Provide implementations for the HolderDetails class and a suitable class to represent transactions. The HolderDetails class is used to store details about the account holder. The Transaction class contains details about past transactions.
8 MARKS

c: Define and implement a new class, called CurrentAccount, derived from Account, that allows withdrawls to proceed up to some overdraft limit.
6 MARKS

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

2.a: Describe the general structure and purpose of the IO Streams classes provided in the Java programming environment. Also describe the mechanism that supports random file access in Java. 5 MARKS

b: Write a Java application that inputs a date as a string in the form 17-07-2018 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 Student class that includes an id number, a name, and course details and a suitable constructor method. Then write a Java program that uses an ArrayList to store a collection of Student 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 Student 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 Student objects. 14 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? 3 MARKS

b: 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. 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. 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. 12 MARKS

4.a: What is the best way to stop executing threads (assuming they still have not finished their work)? Show using a code example how a thread may be created (and started) using an application class that implements the Runnable interface. Assume you have a bank account class that may be accessed by more than one thread of execution simultaneously. Show how the various methods of the class may be made thread safe. 10 MARKS

b: 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 exactly once and there may be multiple producer and consumer threads executing (and attempting to access the buffer) concurrently. 15 MARKS

5: Suppose that you've written a program that displays two messages, as follows:

```
public class NotI18N {
    static public void main(String[] args) {
        System.out.println("Hello.");
        System.out.println("How are you?");
     }
}
```

You then decide that this program needs to display the same or similar messages for people living in France and Spain. Outline the steps needed to properly internationalise this program i.e. the hardcoded English language messages should be removed and replaced with a more flexible mechanism that will facilitate additional language support in the future. 10 MARKS

b: Write a simple GUI-based Java program that may be used to control a washing machine. Use suitable Swing components to allow the washing machine operator to perform the following functions:

1) Switch the machine on.
2) Choose a temperature from a list.
3) Spin speed selection buttons - can be 600, 800 or 122 RPM.
4) Display the current status of the wash cycle.

Show the top-level design of the GUI, including any Panels and related Layout Manager objects that you propose to use. For each of the components you have chosen above, write the code to construct the component, add the component to a container and then set up simple event handling for the component (for those that generate events). The event handlers need only print out a message indicating that they have been called. 15 MARKS