NUI Galway
OÉ Gaillimh

Autumn Examinations 2015

Exam Code(s)	3BCT, 3BP1
Exam(s)	Third Year Computer Science & Information Technology Third Year Electronic and Computer Engineering
Module Code(s)	CT326
Module(s)	Programming III
Paper No.	1
External Examiner(s)	Dr. J. Power
Internal Examiner(s)	Prof. G. Lyons Dr. 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 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. 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: Create a standalone Java application that will count and sum up the number of lines in the text file passed as an argument on the command line. The program should create a **FileReader** object and pass this in the constructor of a **LineNumberReader** object to handle the file reading required.

The **LineNumberReader** class has two useful methods (that could be used):

public String readLine() throws IOException; This method reads a line of text. It returns a String containing the contents of the line, not including any line-termination characters, or null if the end of the stream has been reached.

public int getLineNumber(); This method returns the current line number.

10 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. 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. 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 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. The answer only needs to include source code for the server side application.

10 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 haven't finished their work)? 5 MARKS

b: What is meant by the term *deadlock*? Using the example of the *Dining Philosophers Problem* (covered in class), discuss how deadlock might occur in this case and propose a solution to overcome the problem. 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 exactly once and there may be multiple producer and consumer threads executing (and attempting to access the buffer) concurrently. 10 MARKS

5. Write a Java class called **Rational** for performing arithmetic with fractions. Use integer variables to represent the **private** instance variables of the class - the **numerator** and the **denominator**. Provide a constructor method that enables an object of this class to be initialised when it is declared e.g. the fraction 2/3 would be stored in the object as 2 in the **numerator** and 3 in the **denominator**. 3 MARKS

Provide **public** methods for each of the following:

(a) Addition of two **Rational** numbers. The result should be stored in the target object e.g. if **r1** and **r2** are objects of type **Rational**, calling **r1.add(r2)** would add the value of **r2** to **r1** and then store the new value in **r1**. (Hint: adding 2/3 and 3/4 gives the result 17/12). 4 MARKS

(b) Subtraction of two **Rational** numbers. In the same way as for (a), the result should be stored in the target object. (Hint: subtracting 2/3 from 3/4 gives the result 1/12). 4 MARKS

(c) Multiplication of two **Rational** numbers. In the same way as for (a), the result should be stored in the target object. (Hint: multiplying 2/3 and 3/4 gives the result 6/12). 4 MARKS

(d) Division of two **Rational** numbers. In the same way as for (a), the result should be stored in the target object. (Hint: dividing 2/3 by 3/4 gives the result 8/9). 4 MARKS

(e) Printing **Rational** numbers in the form **a/b**, where **a** is the **numerator** and **b** is the **denominator**. 3 MARKS

Finally, write a suitable driver program that could be used to test your class. 3 MARKS