NUI Galway
OÉ Gaillimh

Spring Examinations 2009 / 2010

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)	Dr. D. Chambers Dr. J. Duggan

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 17/02/2010 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 Bank Account class that includes an account number, holder details, the balance and a suitable constructor method. Then write a Java program that uses an ArrayList to store a collection of Account 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 Account objects based on their balance. Finally, use the version of the Collections.sort() method that allows you to pass your own Comparator object to sort the list of Account objects. 15 MARKS

3.a: Discuss briefly the differences between a process and a thread. How should executing threads be stopped (assuming they still haven't finished their work)? 5 MARKS

b: Show (using simple code examples) how threads may be created (and started) using the following mechanisms:

(i) Application class extends the Thread class.

(ii) Application class 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

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

4.a: 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 Germany. Outline the steps needed to properely 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: Develop 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've chosen above, write the code to construct the component, add the component to a container and then setup 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

5.a: What types of Sockets are supported in the Java networking package and 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 an Integer Object. The server should print out this value and respond to the client with a text based response encapsulated in a String Object. The client should receive the String Object from the server and print out this response. 10 MARKS

c: Implement 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

6. 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**. Also provide a no-argument constructor with default values in case no initialisers are provided.
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