NUI Galway
OÉ Gaillimh

Autumn Examinations 2021-2022

Course Instance Code(s)	3BCT, 3BP
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 Ramona Trestian
Internal Examiner(s)	Prof Michael Madden *Dr Adrian Clear

Instructions: Answer any 4 questions. All questions will be marked equally.

Duration	2 hours
No. of Pages	5
Discipline(s)	Computer Science
Course Co-ordinator(s)	Dr Colm O'Riordan

Requirements:

Release in Exam Venue	Yes [X]	No [ ]
MCQ Answersheet	Yes [ ]	No [X]
Handout	None	
Statistical/ Log Tables	None	
Cambridge Tables	None	
Graph Paper	None	
Log Graph Paper	None	
Other Materials	None	
Graphic material in colour	Yes [ ] No [X]	

Q1:

(a) Implement an Enum in Java called `CouncilTaxBand` which enumerates the following Council Tax bands for property owners based on their property value.

A – above £100,000
B – £100,001 to £150,000
C – £150,001 to £200,000
D – £200,001 to £300,000
E – £300,001 to £500,000

Your Enum class should include a method that takes an integer parameter representing a property value, and returns the band (A, B, C, D, or E) that the value fits within, otherwise `null`.

10 MARKS

(b) Write a `Property` class that includes an address, a value, a number of bedrooms, a number of bathrooms, and a `CouncilTaxBand` as class attributes. The `Property` class should implement the `Comparable` interface to define the natural order for these objects such that the `CouncilTaxBand` is compared first and then the property value.

Write a Java program that uses an `ArrayList` to store a collection of `Property` objects and then sort the list based on natural order.

Also, write the code for a `Comparator` class i.e., a class that implements the `Comparator` interface, that can be used to compare two `Property` objects based on the property value first and then based on the number of rooms.

Finally, use the version of the `Collections.sort()` method that allows you to pass your own Comparator object to re-sort the list of `Property` objects.

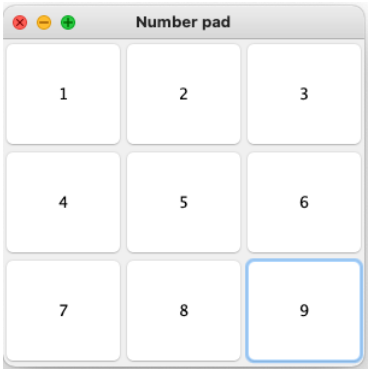
15 MARKS

Q2:

(a) List and describe with the use of diagrams **three** layout managers that can be used to organise components in Swing applications. For each one, provide a code example of how to add a component to a container that has been defined to use that layout manager.

10 MARKS

(b) Write a Java program to produce the GUI below using Swing, which illustrates a Number Pad. When a button is clicked by the user (like button 9 in the figure), it should print the button's number to the console with a message like "Button 9 pressed". The GUI should have a size of 300x300 pixels, a title, and should terminate the application when the close button ('x') is pressed.



15 MARKS

Q3:

(a) The JDK contains two general-purpose List implementations i.e. *ArrayList* and *LinkedList*.
Why is *ArrayList* generally the best performing implementation? Describe the circumstances under which *LinkedList* might offer better performance.

Describe three of the polymorphic algorithms provided in the Java Collections framework. In relation to these algorithms, explain fully the purpose and operation of the following code idiom:

```
int pos = Collections.binarySearch(list, key);
if (pos < 0)
     list.add(-pos-1, key);
```

13 MARKS

(b) Explain using an example what *serialization* is in Java. Your answer should address issues of serializable objects, deserialization, and custom serialization in detail.

12 MARKS

Q4:

(a) Show using a code example how a thread may be created (and started) using an application class that implements the `Runnable` interface.

Include a mechanism in the `Runnable` class to allow it to be shutdown gracefully (i.e., without needing to call the `stop()` method).

Assume you have a bank account class that may be accessed by more than one thread of execution simultaneously. Show how the various business 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 a `String` object. The contents of the `String` may be written 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

Q5:

(a) Explain using examples the difference between Data Sink Streams and Processing Streams.

5 MARKS

(b) Write a Java class called `RAFQ` that uses a `RandomAccessFile` to store diary entries of the format [date][diary entry]. For example, “2022-02-17”“Today is windy!”, where the quotes illustrate an individual UTF string. Your program should include the following two methods for writing a diary entry to and reading a diary entry from the file, respectively:

```
public void writeDiaryEntry(LocalDate timestamp, String diaryEntry)

public String getDiaryEntry(LocalDate timestamp)
```

An example of how the class is intended to be used is as follows. This example would print “Today is dry” to the console:

```
RAFQ raf = new RAFQ("mydiary.txt");
LocalDate timestamp = LocalDate.now();
raf.writeDiaryEntry(timestamp, "Today is sunny");

LocalDate anothertimestamp = LocalDate.parse("2022-01-03");
raf.writeDiaryEntry(anothertimestamp, "Today is dry");

LocalDate andanothertimestamp = LocalDate.parse("2022-04-12");
raf.writeDiaryEntry(andanothertimestamp, "Today is windy!");

System.out.println(raf.getDiaryEntry(anothertimestamp));
```

20 MARKS

END