NUI Galway
OÉ Gaillimh

Semester 1 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)	Professor M. 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 Adrian Clear

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]	

PTO

Q1: Write a Java class called `Book` that has the following class attributes: `String title`, `Author author`, `LocalDate published`, `String isbn`

i. Make it possible to serialize `Book` objects. You should assume that the `Author` class is not serializable. However, you are required to include the `Author` data when serialising `Book` objects. Therefore, your class should indicate that the author attribute is to be ignored as part of default serialisation, and you must implement a suitable `writeObject()` method to perform custom serialisation. It should write all attributes of the Book class to the `ObjectOutputStream` passed to the `writeObject()` method. It should also write the individual attributes of the `author` attribute, which can be obtained from the following methods in the `Author` class:

```
public String getFirstName()
public String getLastName()
public LocalDate getDateOfBirth()
```

7 MARKS

ii. The `Book` class should also include a corresponding `readObject()` method to perform custom deserialization, i.e., it should read all of the `Book` class attributes in the default manner except for the `author` attribute. It should read the relevant strings (first name, last name) and `LocalDate` (date of birth) from the `ObjectInputStream` passed to the `readObject()` method, instantiate an `Author` object, and set it to the `author` class attribute. You can assume an `Author` constructor is available that takes three parameters, one for each of the data read from the stream.

8 MARKS

iii. Write a Java program that creates a list of three `Book` objects. The program should then write the list of `Book` objects to a file using object serialisation. The name of the file should be passed in at the command line to the `main()` method. 5 MARKS

iv. Write another Java program to de-serialise a list of `Book` objects from a file created using serialisation. The name of the file should be passed in at the command line to the `main()` method.

5 MARKS

Q2:

(a) Describe the relationship between a nested inner class and its enclosing class. What is a static nested class and how does it compare to a nested inner class?

5 MARKS

(b) Write a class called `Bank` that maintains a data structure of bank customers of type `Customer`, and their `Account` object. Call this data structure `customers`. You can assume the existence of a `Customer` class. The Bank class should include a nested inner class called `Account`, which has an account number (an `int`) and an account balance (a `double`). Include an `addCustomer` method in the `Bank` class that takes a `Customer` object as a parameter, creates an account for them, and stores the customer and their account in the data structure. The `Bank` class has the responsibility for generating unique account numbers for accounts.

Briefly describe and provide a justification for the data structure you chose for `customers`.

12 MARKS

(c) Assuming the Customer object consists of a First Name (`String`), Last Name (`String`), and a PPS number (`int`), write an appropriate `hashCode()` method for the class based on these three values.

8 MARKS

Q3:

(a) Describe the `Set` interface in the Java Collections Framework. What is its relationship to the `Collection` interface? What are the main characteristics of a `Set` collection? List and briefly describe two classes in the Java Collections Framework that implement the `Set` interface, outlining the difference between them.

6 MARKS

(b) 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);
```

6 MARKS

(c) Write a `Plant` class that includes an ID number, a `genus` name and a `species` name as class attributes. The `Plant` class should implement the

Comparable interface to define the natural order for these objects such that the genus is compared first and then the species.

Write a Java program that uses an ArrayList to store a collection of Plant 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 Plant objects based only on their ID number.

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

13 MARKS

Q4:

(a) The original Lemmings video game is a puzzle game where a player can assign skills to Lemming characters in order to help them navigate through a level. Implement an Enum in Java called Lemming which enumerates the different Lemming skill types from the original Lemmings video game. Include in the enum the speed of each lemming carrying out their respective skill, as follows, Climber=5, Floater=3, Bomber=0, Blocker=0, Builder=2, Basher=4, Miner=3, Digger=1, and a boolean indicating whether or not they are destructive to their environment. Bombers, Bashers, Miners, and Diggers are destructive; the others are not. Provide a suitable toString() method to print information about the enumerated types.

12 MARKS

(b) Write a Java program to provide a GUI for configuring and starting the game of Lemmings. Use suitable Swing components to allow the player to set the following properties and perform the following operations:

1) Start the game.
2) Choose the game resolution from a list (640 x 480; 1280 x 720; 1920 x 1080; 2560 x 1440)
3) Enter a level code (e.g., CAJJMDLJCL)
4) Game difficulty selection buttons (Fun; Tricky; Taxing; Mayhem)
5) Exit the game.

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. The event handlers only need to print out a message to indicate that they have been called.

13 MARKS

Q5:

(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

END