OLLSCOIL NA GAILLIMHE
UNIVERSITY OF GALWAY

Autumn Examinations 2022-2023

Course Instance Code(s)	3BCT1, 3BP1, 3BP4
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. 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]

PTO

Q1:

(a) Explain the difference between threads and processes. Describe with code examples **two** ways of creating and starting a new thread in Java.

10 MARKS

(b) Outline the design and code implementation of a thread-safe Java class for an object that will be used as a buffer to hold an `int` value. After an initial value has been set, 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

Q2.

(a) List and describe the activities involved in the main stages of a test driven development approach.

5 MARKS

(b) You have been asked to write a class to represent a bank account called `Account`. Write the **unit tests** you would use to test the following functionality.

(i) It should be possible to make a deposit of funds to the account as long as the deposit value is positive.

(ii) It should be possible to make a withdrawal from the account as long as there are sufficient funds in the account. If there are insufficient funds in the account, an appropriate exception should be thrown.

(iii) An `Account` is represented with an account number and a balance. It should be possible to serialize an `Account` (including all its member variables) to a file.

15 MARKS

(c) Assume you have a `List` containing `Account` objects, and that the `Account` class has an appropriate `toString()` method implementation. Use a call to the `Iterable forEach` method to print the `Account` objects to the standard output stream.

5 MARKS

PTO

Q3.

Demonstrate using code examples how you would make use of an `Iterator` to count the number of elements in a `Collection` of strings that contain at least one capital letter, lowercase letter, and number.

12 MARKS

Demonstrate using code examples how you would use a `List` to represent the students registered to a module. You should sort the `List` in such a way that the order of students is based on the lexicographic order of their surname followed by their first name.

Illustrate the code for adding students called Steve Higgins and Mary Higgins to the `List` before sorting it.

What would the program output be if you print the `List` to the console?

Note: you can represent a student by their surname and first name only. Your code should include a `toString()` method that represents a `Student` as "[surname], [first name]"

13 MARKS

PTO

Q4:

(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.

Write an appropriate `hashCode()` method for the class based on these three values.

Write an appropriate `equals()` method where two `Plant` instances are considered equal if they have the same ID, species and genus.

Write a `toString()` method that provides an appropriate *formatted* string representation of a `Plant` instance using its three attributes.

13 MARKS

PTO

Q5.

Write a Java class called `Employee` that has the following class attributes: `int idNumber, String name, LocalDate startDate, String jobTitle, float weeklyPay`

(a) The `Employee` class should include a suitable `writeObject()` method, to implement custom object serialisation, that writes out all attributes except the weekly pay to the `ObjectOutputStream` passed to the `writeObject()` method. The weekly pay should instead be appended, in text format, to a CSV (Comma Separated Values) file called payroll.csv in the format "idnumber, weeklypay". Each line of the payroll.csv file will therefore look something like this:

3249930, 420.80

7 MARKS

(b) The `Employee` class should also include a suitable `readObject()` method, that complements the `writeObject()` method, to read all attributes except the weekly pay from the `ObjectInputStream` passed to the `readObject()` method. The weekly pay should instead be read, in text format, from the CSV (Comma Separated Values) file previously created by the `writeObject()` method. The last matching entry, that is an entry with a corresponding idnumber, in the CSV file should be used to set the weeklypay value.

8 MARKS

(c) Write a Java program that creates two employees. The program should then write out the `Employee` objects, using Object Serialisation, to a file before reading the `Employee` objects from the file, again using Object Serialisation.

10 MARKS

END