



Semester 1 Examinations 2019 / 2020

Exam Code(s)	3BCT, 3BP, 3BLE
Exam(s)	Third Year Computer Science & Information Technology Third Year Electronic and Computer Engineering Third Year Electrical and Electronic Engineering
Module Code(s)	CT326
Module(s)	Programming III
Paper No.	1
External Examiner(s)	Dr. J. Howe
Internal Examiner(s)	Prof. M. Madden *Dr. D. Chambers

Instructions: Answer any 4 questions.
All questions carry equal marks.

Duration	2 hrs
No. of Pages	4
Discipline(s)	Computer Science
Course Co-Ordinator	Dr. D. Chambers

Requirements:

Release in Exam Venue	Yes	<input checked="" type="checkbox"/>	No	<input type="checkbox"/>
MCQ Answersheet	Yes	<input type="checkbox"/>	No	<input checked="" type="checkbox"/>
Handout	None			
Statistical/ Log Tables	None			
Cambridge Tables	None			
Graph Paper	None			
Log Graph Paper	None			
Other Materials	None			
Graphic material in colour	Yes	<input type="checkbox"/>	No	<input checked="" type="checkbox"/>

PTO

- 1.a: Implement an Enum in Java which enumerates the months of the year. Include in the enum the relevant position of the month in the calendar i.e. Jan = 1, Feb = 2, etc and the number of days in the month i.e. Jan = 31, Feb = 28, etc. You do not need to provide support for leap years in the implementation. Provide a suitable toString() method to print information about the enumerated types.

12 MARKS

- b: 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

- 2.a: Describe the general structure and purpose of the IO Streams classes provided in the Java programming environment. Also briefly describe the mechanism to support random file access in Java?

5 MARKS

- b: Write a Java application that inputs a date as a string in the form 17-07-2010. 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.

5 MARKS

- c: Develop a simple GUI-based Java program that may be used to control a lighting system. Use suitable Swing components to allow the lights controller to perform the following functions:

- Switch the light system on or off.
- Choose a light intensity level from a list - Level 1, 2, 3 or 4.
- Include Mode of Operation buttons - Manual or Timed.
- A button that, when pressed, will display the current status (i.e. the currently selected settings) of the lights in a text field.

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

- 3.a: Write a network Server program in Java where the Server waits for incoming client connections using stream type sockets. Once a Client connects it sends a String object to the server with a simple query – the server then responds with a text based response. The connection is then terminated. The server should use a separate thread of execution for each new client connection and all interaction between the Server and the Client should be done within this thread. The answer only needs to include source code for the server side application.
12 MARKS
- b: Write another Java application with the same functionality as outlined above, in part a 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. This application does not need to implement a reliable data transfer protocol. The answer only needs to include source code for the server side application.
13 MARKS
4. 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 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. Include the private instance variable joinDate in class Employee to represent when they joined the company. Use the java.util.Date class for this variable.
7 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 wage per hour, so in any given week they will be paid for the number of hours worked in the past week. The class should include a constructor and implement the earnings() method.
5 MARKS
- d: Write a short driver program that creates an array of Employee variables to store references to the various employee objects. In a loop, calculate the payroll for each Employee, and add a €100.00 bonus to the person's payroll if they joined the company over 5 years ago.
8 MARKS

5.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 all their work?

5 MARKS

b: Write a Java animation applet that uses a thread to continuously scroll a message string across the screen from right to left.

8 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.

12 MARKS