



Semester 1 Examinations 2015 / 2016

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. Power
Internal Examiner(s) Prof. G. Lyons
Dr. M. Madden
*Dr. D. Chambers

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

Duration 2 hrs
No. of Pages 4
Department(s) Information Technology

Requirements None

1. The following Java code provides the outline of a simple bank account class:

```
import java.io.*;
public class Account implements Comparable<Account>,
Serializable {
    protected int accnum;
    protected HolderDetails holder;
    protected List<Transaction> transactions;
    protected float balance;

    // Add a suitable constructor here

    // Add methods to make deposits / withdrawals

    // Method to print out account transaction summary

    // Add suitable attribute accessor methods

    // Add method to implement the Comparable interface
}
```

- a: Complete the implementation of the Account class, providing a suitable constructor, attribute accessor methods, methods for making a deposit or withdrawal, a method to print out a transaction summary related to a range of Dates. Also provide an implementation method for the Comparable interface that bases the natural order of these objects on their accnum attribute.
7 MARKS
- b: Provide implementations for the HolderDetails class and the Transaction class. The HolderDetails class is used to store personal details about the account holder. The Transaction class contains details about past transactions including the type of transaction, the amount and the Date.
7 MARKS
- c: Define and implement a new class, called CurrentAccount, derived from Account, that allows withdrawals to proceed up to some overdraft limit. Note that the base Account class shown has no overdraft facility.
5 MARKS
- d: The attributes of class Account are defined as *protected*. What is the implication of this definition?
3 MARKS
- e: What does the statement *implements Serializable* mean? What are the implications of this statement?
3 MARKS

- 2.a: Write the Java code for a `Vehicle` class that implements an accessor method called `getEngineSize()` that returns an `int` representing the engine size for that vehicle e.g. 1895. Then write a Java program that uses an `ArrayList` to store a collection of `Vehicle` 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 `Vehicle` objects based on their engine size. Finally, use the version of the `Collections.sort()` method that allows you to pass your own `Comparator` object to sort the list of `Vehicle` objects. 12 MARKS

- b: The JDK contains two general-purpose `List` implementations i.e. `ArrayList` and `LinkedList`. Which of these classes is generally considered to be the best performing implementation and why is this the case? Outline the circumstances under which each of these classes might offer better performance than the other. Describe 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(l, key);
if (pos < 0)
    l.add(-pos-1, key);
```

13 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 an `Integer` object to the server, the server then responds with a single `String` object. 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 should include full source code for the server application, the client side source code is not required. 13 MARKS

- b: Write another Java application with the same functionality as outlined 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. In this case, only the source code for the client application is required, the server side source code is not required. 12 MARKS

- 4.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 their work)?

5 MARKS

- b: Show, using a simple code example, how a thread may be created (and started) using an application class that 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.

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 exactly once and there may be multiple producer and consumer threads executing (and attempting to access the buffer) concurrently.

12 MARKS

- 5.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 Spain. Outline the steps needed to properly 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: Write 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 1200 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 set up 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