N15/4/COMSC/SP2/ENG/TZ0/XX

Diploma Programme
Programme du diplôme
Programa del Diploma

Computer science
Standard level
Paper 2

Wednesday 18 November 2015 (morning)

1 hour

Instructions to candidates

- Do not open this examination paper until instructed to do so.
- Answer all of the questions from one of the options.
- The maximum mark for this examination paper is **[45 marks]**.

Option	Questions
Option A — Databases	1 – 3
Option B — Modelling and simulation	4 – 6
Option C — Web science	7 – 9
Option D — Object-oriented programming	10 – 13

8815 – 7015

15 pages

Option A — Databases

1. After attending a database course, Paul, the owner of a small shop, decided to create and implement a customer database.

(a) (i) Identify **one** benefit of creating a database. [1]

(ii) Identify **one** cost of the implementation. [1]

Consider this section of his customer database file.

`CustomerTable`

Name	Date	ItemOrdered	QuantityOrdered	InvoiceAmount
Ann Low	18/04/2015	1713	4	200.00
Boris Nicke	18/04/2015	1324	3	180.00
Greta Pink	18/04/2015	1713	3	150.00
Rob Nool	19/04/2015	1648	7	360.00
Ann Low	19/04/2015	1713	5	250.00
Ivor Turk	20/04/2015	1423	6	105.00

(b) (i) State the number of fields that are in each record of the `CustomerTable`. [1]

(ii) Describe, using an example from the table above, why the customer's name is **not** an appropriate primary key. [2]

(iii) Define the term *secondary key*. [1]

(c) Describe the steps in a query that will list all items for which more than five were ordered on 19/04/2015. [4]

Paul finds that the response time to database queries is very slow.

(d) Explain what Paul could do to improve the response time without making changes to his hardware. [5]

(Option A continues on the following page)

(Option A continued)

2. A company lets tourists hire bikes. The data about bikes and hirers is stored in a database file. The following table shows this data for one day.

BikeHireTable

HirerID	HName	HPhone	TimeOut	TimeIn	BikeID	BikeMake	BikeModel
BL567	Boris Lok	99123456	09:00	11:00	DU12	BMS_11	A
CL167	Ivy Lok	93123455	09:00	11:00	DU14	AVG_00	B
AL751	Ann Summer	43453657	09:00	17:00	DU54	AVG_00	A
FC345	Fred Cohen	38321432	10:00	15:00	DU23	XYZ_94	C
…	…	…	…	…	…	…	…

The structure of the table can be summarized using the following **shorthand notation**.

BikeHireTable
(<u>HirerID</u>, HName, HPhone, TimeOut, TimeIn, <u>BikeID</u>, BikeMake, BikeModel)

(a) Identify **three** functions of a database management system in an application such as hiring bikes. [3]

There are several aspects of the BikeHireTable that are unsatisfactory and it is decided to normalize this database.

(b) Outline the purpose of normalization. [2]

(c) State the characteristics of

(i) 1st Normal form (1NF); [1]

(ii) 2nd Normal form (2NF); [1]

(iii) 3rd Normal form (3NF). [1]

(d) Construct the database to 3NF. Using the **shorthand notation**, clearly show the structure of the database in the **1st**, **2nd** and **3rd** normal forms. [7]

(Option A continues on the following page)

(Option A continued)

3. (a) Distinguish between the logical and the physical design of a database. [4]

(b) Identify **three** items of information that a data dictionary contains. [3]

(c) Explain **two** responsibilities of a database administrator. [4]

An online information service provides a database of job openings and also posts users' details. Users can view job listings and reply electronically with their details to the jobs that interest them.

(d) Discuss **one** advantage and **one** disadvantage of using a database in this online information service. [4]

End of option A

Option B — Modelling and simulation

4. Air pollution is a concern for public health. One measure of air pollution is the amount of PM10 (particles measuring less than 10 micrometres) in the air. European legislation has been put into place which requires that any town with over 100 000 inhabitants has to keep within the following limits of PM10:

1. The average PM10 value in one day must not exceed 50 micrograms per cubic metre (50 $\mu g\,m^{-3}$) more than 35 times in the year.
2. The daily values averaged across the whole year (measured from 1 January) must not exceed 40 micrograms per cubic metre (40 $\mu g\,m^{-3}$).

In an industrial town, PM10 measurements are taken once an hour, recorded and stored. A complete model of hourly measurements is built up over the year. At the end of the year the town produces a report which includes the following:

- a chart of the average daily values
- the number of days the average exceeded 50 $\mu g\,m^{-3}$
- the daily values averaged across the year.

(a) Outline, using a sketch or otherwise, a method of organizing the stored data in a spreadsheet for easy reference and analysis. [3]

Each day the average value from the previous day is published in the local press, together with the number of days, if any, that the daily limit (50 $\mu g\,m^{-3}$) has been exceeded since 1 January.

(b) Describe how the hourly results could be used to output the results required daily and also to produce an end-of-year report. [5]

The model can be used to produce data on patterns of pollution over time.

(c) (i) Identify **two** of these patterns. [2]

(ii) Explain how one of the patterns from part (c)(i) could help the town plan for the future. [2]

At one point in the year, the town's daily values, averaged since 1 January, are 45 $\mu g\,m^{-3}$. A "no car" weekend is introduced, in which people are not allowed to drive their cars within the town. It successfully reduces the average PM10 count but has a negative effect on local commerce.

(d) Explain how the model could be used to estimate the minimum number of "no car" weekends that would be needed to keep the town's daily values averaged across the year from exceeding 40 $\mu g\,m^{-3}$. [5]

(Option B continues on the following page)

(Option B continued)

5. A game for very young children is played on a board, which is represented by a 4×4 square. The game requires two players

The aim of the game is to be the first player to reach HOME which is the bottom right square. The players take turns to move between squares.

Players can move to an adjacent square which, from their current position, is:

- vertically down
- horizontally to the right
- diagonally in the direction of HOME.

If one of the moves is blocked by the other player, the player can jump over the other player and land in the square on the opposite side.

In the following diagram, player A is in square 1 and player B is in square 6.

A	2	3	4
5	**B**	7	8
9	10	11	12
13	14	15	HOME

(a) List the possible squares to which player A is able to move. [1]

(Option B continues on the following page)

(Option B, question 5 continued)

(b) Player A moves to square 2 and then player B moves to square 10.

1	**A**	3	4
5	6	7	8
9	**B**	11	12
13	14	15	HOME

For each of the moves that player A **could now take**, suggest and justify the best move that player B **could then take**. [3]

The game is simulated by a computer program.

(c) (i) Outline **one** way of representing player A and player B, and their positions, in memory. [2]

(ii) Using your structure from part (c)(i), state the current positions of player A and player B, as shown in the diagram in part (b). [1]

(iii) Using your answer to part (c)(i), outline how each **possible** move from the player's current position could be identified. [2]

(iv) Suggest how a **best move** could be selected from the possible moves. [3]

The computer will always attempt to find the best path to the HOME square.

(d) Outline the effect that changing to a much larger board would have on running the game on a computer. [4]

(Option B continues on the following page)

(Option B continued)

6. A self-driving car relies on a series of lasers and cameras that combine with high definition maps to give a three-dimensional (3D) view of the environment. Rules are programmed into the system so that the car drives at an optimum speed, keeps a safe distance from any vehicle in front, uses fuel economically and follows the traffic regulations.

(a) Outline the need for high definition maps held in memory. [2]

(b) Outline how analysing successive 3D images in memory helps the car to drive safely. [3]

(c) Outline the time and memory needs of 3D visualization in this situation. [3]

(d) Discuss the social implications of self-driving cars. [4]

End of option B

Option C — Web science

7. The code below is part of an XML (*Extensible Mark-up Language*) document that contains details of a DVD collection.

```
<collection>
  <dvd>
    <title>The Hobbit</title>
    <genre>Fantasy</genre>
  <dvd>
    <title>Sleepless in Seattle</title>
    <genre>Romance</genre>
  </dvd>
<!--more DVDs entered here-->
</collection>
```

(a) Identify the error in the section of XML code shown above. [1]

In some applications, XML is used instead of HTML principally because of its extensibility property.

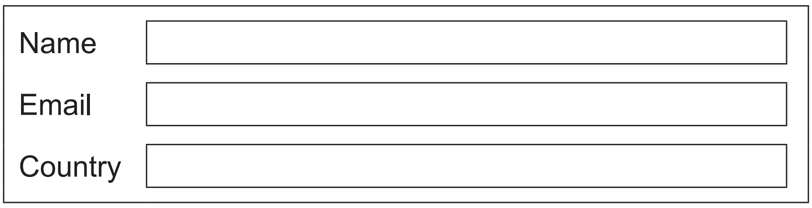
(b) Outline the meaning of this property in the context of XML code. [2]

(c) Describe **one** benefit of storing HTML formatting information in a CSS file. [2]

An XML document contains details of a person's DVD collection. The XML code will be sent to the web server using the internet protocol suite, which includes the TCP and IP protocols.

(d) Describe how these two protocols work together when sending data over the internet. [4]

The following web form is part of a login page for an online shopping site.



(e) (i) Explain how client-side scripting might be used on this login page before the page is sent to the web server of the shopping site. You should make reference to any software used. [3]

(ii) Outline the benefit of using client-side scripting for the shopping site. [2]

A social networking site stores multiple profile photographs of each client on its web server.

(f) Explain how the use of scripts allows the user to change their profile photograph without reloading the complete page. [4]

(Option C continues on the following page)

Turn over

(Option C continued)

8. When a user requests a file from a particular website, the website uses lossy compression to send the file to the user over the internet.

(a) Discuss how this use of lossy compression might affect the user's experience. [5]

The three essential processes of a web search engine are:
- crawling
- indexing
- searching.

(b) Outline the functions of each of these three processes. [6]

(c) Explain why the PageRank algorithm might discriminate against new websites. [3]

(d) Explain how a search engine is able to maintain an up-to-date index when the web is continually expanding. [3]

9. A large online retail company is relocating to a new purpose-built site and is considering incorporating cloud computing into its IT strategy.

(a) Describe **two** architectural features that are essential components of a sustainable public cloud computing model. [4]

In making decisions regarding their computing facilities, the company has to consider the following factors:
- the storage of sensitive financial data
- the extensive use of email for promotional purposes
- the increased demand over the Christmas period
- any upcoming development projects.

Many companies now use a hybrid approach to cloud computing in which they make use of both private and public clouds.

(b) Explain how this particular company might use a hybrid approach. [6]

End of option C

Option D — Object-oriented programming

A small health clinic with three doctors operates in a village. All clients of the clinic have their details stored in the clinic's database. Patients that visit the clinic during the day are given a priority rating (1–3) and are seated in a waiting room to wait for the next available doctor. When it is their turn, the patients are taken from the waiting room to have a consultation with their assigned doctor, who makes a diagnosis, provides treatment and writes a prescription.

The clinic's system is coded in Java. There are many objects in this system and some of them are listed below.

Object	Description
`Doctor`	A licensed professional who treats patients in the clinic.
`Patient`	A sick person who requires a consultation with a doctor.
`WaitingRoom`	A place where patients wait for their consultations.
`Consultation`	A dated meeting between a doctor and a patient which results in a diagnosis, treatment and a prescription for medication.
`Treatment`	A dated record of all actions and medication prescribed to treat the patient's diagnosed condition.

The three objects `Patient`, `WaitingRoom` and `Treatment` have been defined in the following UML diagrams:

```
Patient
-----------------------------------
Integer id
String name
Integer priority
String doctor
-----------------------------------
setId (Integer id)
setName (String name)
setPriority (Integer priority)
setDoctor (String doctor)
Integer getID()
String getName()
Integer getPriority()
String getDoctor()
String toString()
```

```
WaitingRoom
-----------------------------------
Patient[10]patients
-----------------------------------
add(Patient newPatient)
void callNextPatient()
Integer findNextPatientIndex()
remove(Integer n)
```

```
Treatment
-----------------------------------
String date
Integer patientId
String doctor
String actions
String medication
-----------------------------------
setDate (String date)
setPatientId (Integer id)
setDoctor (String doctor)
setActions (String actions)
setMedication (String medication)
String getDate()
Integer getPatientID()
String getDoctor()
String getActions()
String getMedication()
String toString()
```

(Option D continues on the following page)

(Option D continued)

The `Patient` and `WaitingRoom` objects are implemented as follows:

```
public class Patient
{
  private int id;
  private String name;
  private int priority;
  private String doctor;

  public Patient(int i, String n, int p)
  {
    id = i;
    name = n;
    priority = p;
    doctor = null;
  }
  public void setId(int i) { id = i; }
  public void setName(String n) { name = n; }
  public void setPriority(int p) { priority = p; }
  public void setDoctor(String d) { doctor = d; }
  public int getId() { return id; }
  public String getName() { return name; }
  public int getPriority() { return priority; }
  public String getDoctor() { return doctor; }
  public String toString() { return id+" "+name+" "+priority+" "+doctor; }
}

public class WaitingRoom
{
  private Patient[] patients = new Patients[10];

  // uses default constructor

  public void add(Patient newPatient)
  // adds the new patient in the next empty array location
  {
    int i = 0;
    while ((patients[i] != null) && (i < 10))
    {
      i=i+1;
    }
    if (i==10) { System.out.println("No more space in the waiting room."); }
    else { patients[i] = newPatient; }
  }
```

(Option D continues on the following page)

(Option D continued)

```
  public void callNextPatient()
  // finds the next patient, outputs their details
  // and removes the patient from the array
  {
    int index = 0;
    if (patients[0]==null)
    {
      System.out.println("The waiting room is empty.");
    }
    else
    {
      index = findNextPatientIndex();
      remove(index);
    }
  }

  private int findNextPatientIndex()
  // returns the index of the first patient with the
  // highest priority in the array patients
  {
    int max = 0;
    //... code missing ...
    return max;
  }

  private void remove(int n)
  // outputs the data of the patient instance at array index n
  // and removes that patient by shifting all remaining patients
  // by one index towards the front of the array
  {
    //... code missing ...
  }
}
```

(Option D continues on the following page)

(Option D continued)

10. (a) Define the term *constructor*, using an example from the code on pages 12 and 13. [2]

(b) Describe **one** additional field that might have been included in the `Patient` class. Include a data type and sample data in your answer. [2]

(c) Describe the relationship between the `Patient` object and the `WaitingRoom` object. [2]

Consider the `WaitingRoom` class as presented on pages 12 and 13.

(d) Construct the missing lines of code in the `findNextPatientIndex()` method to return the index of the first patient with the highest priority in the `patients` array.
Note: the highest possible priority is 3. [3]

(e) Construct the `remove(int n)` method which outputs the data of the patient object at index `n` and then removes that patient object by moving all remaining patient objects one index towards the front of the `patients` array.
You may assume that `n` is a valid index between 0 and 9, and that an instance of `Patient` exists at that index. [6]

11. (a) In relation to the `Patient` class, outline **one** advantage of encapsulation. [2]

(b) In relation to the `Treatment` object, discuss **one** ethical consideration when designing software that stores patients and their illnesses. [4]

The clinic would like to start storing details in a `Doctor` object, including full name, telephone number and whether the doctor is present or not. For example:

name:	Dr Henriëtte Mănescu-Rața
phone:	0734511122
present:	true

(c) Design the `Doctor` object using a UML diagram. [3]

(d) In relation to the `Doctor` object, outline the need for extended character sets as used by modern programming languages. [3]

(Option D continues on the following page)

(Option D continued)

12. `Treatment` objects are being instantiated throughout the day and added to a collection. The object `treatmentFile` contains the following methods which act on that collection:

- `getNext()` which reads the next treatment from the collection and returns it
- `hasNext()` which returns false when there are no more treatments in the collection.

Construct the method `showMedicationByDoctor()`, which will take the name of a doctor as a parameter and output the medication for each treatment in the collection that has been provided by that doctor. You may assume that `treatmentFile` has been declared as a global variable, that it is open for reading, and that the first time `getNext()` is called it will return the first treatment from the collection. [6]

13. The `Treatment` object needs to be developed further. There are three possible types of treatment and this is to now be recorded.

- `ambulatory` – the patient is treated and goes home afterwards
- `in-patient` – the patient spends one or more nights in the clinic
- `referral` – the patient is sent to a hospital in a nearby city.

All treatments have common fields such as ID of the patient, date and a doctor object, but other fields are different. For example, `ambulatory` and `in-patient` treatments include medication while `referral` does not. On the other hand, `referral` includes the name of the hospital that the patient was sent to and whether or not ambulance transportation was used. `In-patient` treatment includes a room number.

(a) Construct diagrams to show how inheritance can be used to re-design the `Treatment` class. [6]

(b) Describe **three** advantages of modularity in program development. [6]

End of option D