International Baccalaureate®
Baccalauréat International
Bachillerato Internacional

COMPUTER SCIENCE
STANDARD LEVEL
PAPER 2

Tuesday 18 November 2014 (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	10–12

Option A — Databases

1. (a) (i) Describe, with the use of an example, the difference between an operation and a transaction. *[3]*

(ii) Identify a reason why a transaction may need to be rolled back. *[1]*

(iii) State the effect of rolling back a transaction. *[1]*

A database is to be used by a small bank to process deposits, withdrawals, account balance enquiries and the creation of new accounts.

(b) Outline **two** reasons for using a database for this application. *[4]*

(c) Explain, with examples, how this application might use data validation and data verification in processing a deposit. *[4]*

(d) Discuss, with an example, the importance of isolation (the ACID property) in this application. *[3]*

2. A company sells handmade hats online. They plan to introduce a database management system (DBMS) which is to be used as part of their online shop. The data the company stores for each hat are: size, price and the name of the person who made the hat.

(a) Identify the data types for the data of each hat. *[3]*

(b) Identify **three** capabilities that must be supported by DBMS tools. *[3]*

One of the features of a DBMS is security.

(c) For the scenario above, discuss **two** security features needed for this DBMS. *[4]*

(d) Outline the nature of a data dictionary. *[2]*

(e) Explain the importance of data modelling in the design of a database. *[3]*

(Option A continues on the following page)

(Option A continued)

3. A bookstore has hired you to design a database for keeping track of their books. They have provided the following description:

- each **book** is entered into the database with a title, an author, a publication date, a list price and the number of copies in the bookstore
- each **bookcase** contains shelves and has an aisle number and a bookcase number (*eg* bookcase 23 has 5 shelves and is on aisle 4)
- each **shelf** contains book titles
- a **sales log** contains a book title, the customer name, the price paid and the date of the sale.

(a) Describe the characteristics of a normalized database. *[2]*

(b) Outline the issues caused by redundant data. *[2]*

(c) Construct an entity-relationship diagram (ERD) for the bookstore's database that exactly represents the information in the four bullet points shown above. *[3]*

(d) (i) Identify a component of the ERD that is not consistent with 3rd Normal Form (3NF). *[1]*

(ii) Outline what could be done to transform the ERD into 3NF. *[3]*

(e) Discuss **one** issue related to privacy that might be raised if the data in this database were accessed by the police. *[3]*

End of Option A

Option B — Modelling and simulation

4. The amount of an antibiotic medicine given to a patient in each dose depends on the weight, in kg, and the age of the patient at the start of the treatment.

The amount to be given to different patients is calculated as follows.

Age	**Amount of medicine per kg**
18 – 40	1.50 mg
41 – 60	1.75 mg
over 60	2 mg

(a) Calculate the amount in each dose to be given to a patient who is 20 years old and weighs 50 kg. *[1]*

(b) State the variables and calculation needed to model the amount required for a patient. *[2]*

Consider the case where the medicine must be given every 8 hours for a period of 10 days.

(c) Construct an algorithm that uses this model to create a list of days, times and amount in each dose to be taken by a patient, assuming that the first dose is given at 8:00 on the morning of day1. The list should refer to the dates as day1, day2 *etc* and the times should be in 24-hour format. *[4]*

Over time the patient's body absorbs some of the medicine. Immediately **before** the next treatment the amount of medicine in the body is 20% of that which was present immediately **after** the last treatment. A record needs to be kept of the amount of medicine which is in the body at each stage of the treatment.

(d) (i) With reference to any software with which you are familiar, explain how a model could be constructed to calculate the amount of medicine in the blood of a particular patient immediately **after** each amount is given. *[4]*

(ii) Identify the effect of input errors that could occur in this model. *[1]*

(iii) Suggest a suitable way of checking the accuracy of the model. *[3]*

(Option B continues on the following page)

(Option B continued)

5. A new town has experienced population growth of 7 % over the past year, which is more than was expected. At present there are 25 000 inhabitants. For the town planner, it is important to predict the future population size in the town.

(a) State how a simulation could be set up to predict the population over the next 10 years using only the current rate of growth. *[1]*

(b) Identify **three** factors that affect population growth in a town. *[3]*

(c) Outline an improvement to the simulation that could be made using the three factors you identified in part (b). *[3]*

(d) Discuss the accuracy of simulating population growth in this way. *[4]*

(e) Predict **two** social consequences of an inaccurate simulation of population growth in a town. *[4]*

6. Computer-aided design incorporates 2D and 3D visualization. A car manufacturer decides to adapt a current design to include extra features. The designer starts by loading a wire frame version of the current design.

(a) Outline the way in which the wire frame graphic represents the car design. *[2]*

(b) (i) Identify **one** change to the design of the car that could be made using the wire frame. *[1]*

(ii) Outline how the wire frame could be manipulated to make this change. *[2]*

(iii) Outline how the designer could interact with the computer in order to achieve this. *[2]*

After changes have been made to the basic design, changes to the interior and exterior of the car can be made.

(c) Identify **two** changes to the interior and exterior that the designer may want to make. *[2]*

To make these changes the designer needs a different view.

(d) Describe the visualization techniques that would be applied to convert the wire frame to a 3D view of the car, as it would appear when manufactured. *[6]*

End of Option B

Option C — Web science

7. A student in the United Kingdom is viewing a page from a newspaper's website based in South Africa.

(a) Using this example, distinguish between the internet and the World Wide Web. *[3]*

Many newspapers now host an internet version through which users can read the various news stories.

(b) Identify **two** other electronic ways in which newspapers provide information through the use of the technology brought about by the evolving web. *[2]*

The uniform resource locator (URL) is used to identify a specific internet resource. An example from an international newspaper is

http://www.southafricantimes.com/football/mon/rt

(c) By using the URL example given above, identify **three** characteristics of a URL. *[3]*

An HTML page from a particular website displays a form containing radio buttons. These buttons allow the user to choose a particular option. This option will determine the content of the next web page to be delivered from the website's server.

(d) With direct reference to the common gateway interface (CGI), explain how this process takes place. *[4]*

(e) As the number of websites continues to increase, the range of addresses also needs to increase. Suggest, with reasons, which protocol has been updated to allow this to happen. *[3]*

(Option C continues on the following page)

(Option C continued)

8. Two new scientific magazines have been made available online.

(a) By making reference to both content **and** links, describe how a search engine using the HITS algorithm might produce different page rankings for the two magazines. *[4]*

Both magazines allow content to be downloaded. This content could be text or images, for example.

(b) Outline why different compression techniques might be used for transferring this information. *[4]*

(c) Explain, with an example, how the same search terms on the same search engine might produce different results for different users. *[3]*

(d) Identify **two** different types of distributed systems, giving an example of how each one might be used. *[4]*

9. Laptops that only use cloud computing for all services and applications are now being marketed.

(a) (i) Identify **two** features of this type of laptop relating to cloud computing. *[2]*

(ii) Outline **two** possible disadvantages to the user of this type of laptop. *[4]*

(b) By looking at **one** positive and **one** negative aspect, discuss the effect of the lack of central control of the web. *[5]*

(c) For **one** example of intellectual property, evaluate **one** measure that has been taken for its protection on the web. *[4]*

End of Option C

Blank page

Option D — Object-oriented programming

A delivery company uses trains in its operations. It uses an object-oriented program to keep track of its trains and the parcels that it carries.

The company has many objects in their program; here are some of them.

Object	Description
`Train`	Each `Train` is made up of `RollingStock` objects, each of which is either a `Wagon` or an `Engine`.
`RollingStock`	A `RollingStock` object can be an `Engine` (that can pull) or a `Wagon` (that needs to be pulled). Each `RollingStock` has a unique ID number and a weight.
`Engine`	A variety of `RollingStock`. Each `Engine` has a maximum weight that it can pull.
`Wagon`	A variety of `RollingStock`. Each `Wagon` has a maximum cargo weight.
`Parcel`	Each `Parcel` is tagged with a tracking number, the addresses from where it came (origin) and to where it is going (destination) and its weight.

The code on the following pages implements the `Train` class used in this program.

(Option D continues on the following page)

(Option D continued)

```
public class Train
{
  private Engine[] mEngines;
  private Wagon[] mWagons;
  private int mEngineCount;
  private int mWagonCount;
  private int mTrainNumber;
  private double mWeight;       //  Total weight in kilograms
  public Train(int number)
  {
    mTrainNumber = number;
    mEngines = new Engine[6];  // The train can have up to 6 engines
    mEngineCount = 0;
    mWagons = new Wagon[100];  // The train can have up to 100 wagons
    mWagonCount = 0;
    mWeight = 0;
  }
  public void addEngine( Engine newEngine )
  {
    mEngines[mEngineCount] = newEngine;
    mEngineCount++;
  }
  public Engine removeEngine()
  {
    mEngineCount--;
    return mEngines[mEngineCount];
  }
  public void addWagon( Wagon newWagon )
  {
    mWagons[mWagonCount] = newWagon;
    mWagonCount++;
  }
  public Wagon removeWagon()
  {// Code to be written
  }
  public double getWeight()
  {// Code to be written
  }
  ...
}

public class RollingStock
{
  private int mIDNumber;
  private double mWeight;
  public RollingStock(int ID, double weight)
  {
    mIDNumber = ID;
    mWeight = weight;  // Weight is in kilograms
  }
  // Accessor methods
  public double getWeight() { return mWeight; }
  public int getID() { return mIDNumber; }
  ...
  // Other methods
     ...
}
```

(Option D continues on the following page)

(Option D continued)

```
public class Engine extends RollingStock
{
  private double mPullingWeight;     // maximum weight engine can pull
  public Engine(int ID)
  {
    super(ID, 120000);               // Engines weigh 120000 kilograms
    mPullingWeight = 1400000;        // Engines can pull 1400000 kilograms
  }
  // Accessor methods
  public double getWeight() { return super.getWeight(); }
  ...
  // Other methods
    ...
}

public class Wagon extends RollingStock
{
  private Parcel[] mParcels;
  private int mParcelCount;
  public Wagon(int ID)
  {
    super(ID, 32000);         // Empty wagon weighs 32000 kilograms
    mParcels = new Parcel[100];
    mParcelCount = 0;
  }
  // Accessor methods
  public int getWagonID() { return this.getID(); }
  public double getWeight()
  {
    // Code to be written
  }
  ...
  // Other methods
  ...
}
```

10. (a) Define the function of a *constructor*. *[2]*

(b) Outline the advantages of polymorphism, using the `RollingStock` class as an example. *[3]*

(c) Construct a unified modelling language (UML) diagram of the `Train` class. *[3]*

(d) Construct a method `getNumberOfWagons()`, part of the `Train` class, that returns the number of wagons currently coupled to the train. *[2]*

(e) Construct the `removeWagon()` method that will remove one wagon from a train and return the removed object. Include appropriate error checking. *[5]*

(Option D continues on the following page)

(Option D continued)

11. (a) Outline **one** advantage of using standard library collections. *[2]*

(b) Describe **two** ways in which programming by a team differs from programming by an individual working alone. *[4]*

The following code implements the `Parcel` class used in the delivery company's program.

```
public class Parcel
{
  private int trackingID;
  private double weight;
  public String destinationAddress;
  public String originAddress;
  public Parcel(int ID)
  {
    trackingID = ID;
    weight = 0;
  }
  public void setWeight(double newWeight) { weight = newWeight; }
  public double getWeight() { return weight; }
}
```

The origin and destination addresses are stored in a `Parcel` object as simple strings. However, addresses are complex and there are a lot of different pieces of information that may or may not be present such as a first name or a business name, in addition to house number, street name, city and country.

It has been decided to create a new `Address` class to handle this information.

(c) State the appropriate data type to be used in the `Address` class to store

(i) the street name; *[1]*

(ii) the building number; *[1]*

(iii) an indication of whether or not this is a business address. *[1]*

(d) Identify the changes to the `Parcel` class that will be needed to make use of the new `Address` class. *[3]*

Separate `OriginAddress` and `DestinationAddress` classes will be created. The destination address may contain special instructions to the delivery person. The origin address contains a variable that indicates if the parcel was collected from the customer's house or from the local post office.

(e) Outline how these **two** new classes can be created with minimal duplication of code. *[3]*

(Option D continues on the following page)

(Option D continued)

12. (a) Consider the following code fragment.

```
Train A = new Train(123);
Engine B = new Engine(7);
A.addEngine(B);
Wagon C = new Wagon(23);
A.addWagon(C);
Wagon D = new Wagon(66);
A.addWagon(D);
Wagon E = new Wagon(71);
A.addWagon(E);
A.addEngine(new Engine(9));
```

(i) Draw the `mEngines` array after the code fragment has been executed. *[2]*

(ii) State the value of `mEngineCount` after the code fragment has been executed. *[1]*

(iii) Draw the `mWagons` array after both the code fragment above **and** the code fragment below have been executed. *[2]*

```
Wagon F = A.removeWagon();
F = A.removeWagon();
A.addWagon(new Wagon(214));
```

The parcels loaded into a wagon cannot weigh more than the capacity of the wagon. A train's engines must have enough combined power to pull the loaded wagons. The company needs to be able to check that these requirements are being met.

(b) Construct the `getWeight()` method in the `Wagon` class that returns the total **combined** weight of the parcels currently in the wagon **and** the wagon itself. *[4]*

(c) Construct the `getWeight()` method in the `Train` class that returns the total **combined** weight of all the parcels, engines and wagons in a train. *[4]*

(d) Explain why having a `getWeight()` method in both the `Train` and `Wagon` classes does not cause a compiler error, even though the `Train` class does not inherit from the `RollingStock` class. *[2]*

End of Option D