INTERNATIONAL SECONDARY CERTIFICATE EXAMINATION
NOVEMBER 2023

COMPUTER SCIENCE: PAPER I

Time: 3 hours 150 marks

PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY

1. This question paper consists of 16 pages. Please check that your question paper is complete.

2. This question paper is to be answered using object-oriented programming principles. Your program must make sensible use of methods and parameters.

3. This paper is divided into two sections. All candidates must answer both sections.

4. This paper is set in programming terms that are not specific to any particular programming language (Java/Delphi) or database (Access/MySQL/JavaDB).

5. Ensure that you answer the questions in the manner described because marks will be awarded for your solution according to the specifications that are given in the question.

6. Only answer what is asked in each question. For example, if the question does not ask for data validation, then no marks are awarded for it; therefore, no code needs to be written for data validation.

7. If you cannot get a section of code to work, comment it out so that it will not be executed and you can continue with the examination. If possible, try to explain the error to aid the marker.

8. Your programs must be coded so that they will work with any data, not just the sample data supplied or any data extracts that appear in the question paper. You are advised to look at the provided data files carefully.

9. Make sure that routines such as searches, sorts and selections for arrays are developed from first principles and that you do not use the built-in features of a programming language for any of these routines.

10. All data structures must be defined and declared by you, the programmer. You may not use the interface's components to store and retrieve data.

11. Read the whole question paper before you choose a data structure. You may find that there could be an alternative method of representing the data that will be more efficient considering the questions that are asked in the paper.

12. You must save all your work regularly on the disk you have been given, or the disk space allocated to you for this examination. You should also create a backup of the original files before you start in case your solution accidentally modifies the original version.

13. If your examination is interrupted by a technical problem such as a power failure, you will, when you resume writing, be given only the time that was remaining when the interruption began, to complete your examination. No extra time will be given to catch up on work that was not saved.

14. Make sure that your examination number appears as a comment in every program you code and on every printed page you hand in.

15. Check every printed page you print, making sure you submit the code for every question.

16. Check that your printout has correct indenting and formatting and has not truncated any text.

17. Check that each page is clearly legible with a font size of 12.

18. Print a code listing of all the programs/classes/output files you code. Printing must be done after the examination. You will be given half an hour to print after the examination is finished. Your teacher will tell you what arrangements have been made to print your work.

19. You should have the following two folders (in bold) and files. These files are to be used as data for this examination. Note that the database files are in MS Access, JavaDB and MySQL format. Ensure that you can open the files with the packages you will use to code your solutions to this examination.

    Section A:
    ChampionsForChange_Access.mdb
    ChampionsForChange _JavaDB.sql
    ChampionsForChange _MySQL.sql
    SQLAnswerSheet.rtf
    SQLBrowser.exe

    Section B:
    Members.txt
    Sponsors.txt

SCENARIO:

Champions For Change is an organisation that runs community centres for underprivileged children in cities in Botswana and Namibia. The centres rely on donations for operating costs and maintain a list of donors. Donors can be private individuals or businesses; all receive a tax certificate for every donation. Each centre employs many staff to assist the children who attend. The centres try to have at least 1 staff member for every 30 children who attend centre activities.

SECTION A SQL

QUESTION 1

All the data on centres run by Champions For Change, the donors and donation records are stored in a database named **ChampionsForChange**.

tblCentres contains the details of all the centres run by Champions For Change.

FIELDS	DATA TYPE	DESCRIPTION
CentreID	INTEGER	A unique auto-numbered identification number for each centre.
CentreName	TEXT	The name of the centre.
City	TEXT	The city the centre is located in.
NumStaff	INTEGER	The number of staff working at the centre.
NumChildren	INTEGER	The number of children attending the centre.

The first 10 records of tblCentres:

CentreID	CentreName	City	NumStaff	NumChildren
1	Foot Forward Three	Windhoek	7	214
2	Strive for Success	Serowe	4	143
3	Soaring Eagles	Henties Bay	6	138
4	Flight of Champions	Selebi Phikwe	4	67
5	Towards Greatness	Maun	10	189
6	Guardians for Hope	Francistown	3	53
7	Champions in the Mix	Swakopmund	7	132
8	Victory is Ours	Francistown	9	269
9	Foot Forward Two	Windhoek	9	275
10	On the Path	Gaborone	6	131

tblDonors contains the details of the donors registered with Champions For Change.

FIELDS	DATA TYPE	DESCRIPTION
DonorID	INTEGER	A unique auto-numbered identification number for each donor.
DonorName	TEXT	The name of the donor.
Contact	TEXT	The email address of the donor.
Individual	BOOLEAN	Whether the donor is a private individual.
TaxNum	TEXT	The tax number of the donor.

The first 10 records of tblDonors:

DonorID	DonorName	Contact	Individual	TaxNum
1	Matthews Peterson	mpeterson@gmail.com	TRUE	VCI8026324513
2	Shania Julien	shanjulien@gmail.com	TRUE	EOM7692719782
3	Simply Banking	finance@simplybanking.com	FALSE	OHA3715291767
4	Green Meadows	donations@greenmead.com	FALSE	RVD1008791134
5	Gweneth Lebrun	gwenethl@gmail.com	TRUE	NWZ5834779217
6	Luxury Cars	mantashe@luxurycars.com	FALSE	BCE8772983995
7	Botswana Bank	dhlomos@bb.com	FALSE	IJD5582157342
8	Valery Read	valery1232@yaho.com	TRUE	VEL4238920271
9	Getaway Travels	henningk@getawayadventures.com	FALSE	CKR6744625891
10	Yamikani Janz	janzfamily@gmail.com	TRUE	VXE4890363135

tblDonations contains the details of all donations received by Champions For Change

FIELDS	DATA TYPE	DESCRIPTION
DonationID	INTEGER	A unique auto-numbered identification number for each donation.
DonorID	INTEGER	The ID of the donor who donated. This is a foreign key to tblDonors.
CentreID	INTEGER	The ID of the centre that received the donation. This is a foreign key to tblCentres.
Amount	INTEGER	The donation amount. This is an integer value.
CertificateIssued	BOOLEAN	Whether a tax certificate has been issued for the donation.
DonationDate	DATE	The donation date.

The first 10 records of tblDonations:

DonationID	DonorID	CentreID	Amount	CertificatIssued	DonationDate
1	1	3	19965	FALSE	2020/01/04
2	22	25	32668	FALSE	2020/01/09
3	17	25	42536	TRUE	2020/01/16
4	18	1	95615	TRUE	2020/01/24
5	5	22	54583	FALSE	2020/02/09
6	6	9	28902	TRUE	2020/02/13
7	6	21	60880	TRUE	2020/02/19
8	2	8	93254	TRUE	2020/04/04
9	9	10	80132	FALSE	2020/04/04
10	25	18	81258	FALSE	2020/05/24

1.1 Display the details of the donors who are private individuals. Sort the output alphabetically according to the name of the donor. Sample output is given below:

Note only the first six records are shown.

DonorID	DonorName	Contact	Individual	TaxNo
12	Alexander Rodriguez	alex@rodriguez.com	TRUE	FAQ8335827303
16	Burton Shin	shinbone@yaho.com	TRUE	CPT6400551735
19	Dara Magee	dmagee@aol.com	TRUE	GYA0523058403
5	Gweneth Lebrun	gwenethl@gmail.com	TRUE	NWZ5834779217
14	Hemming Sokolovsky	sokolovskyh@aol.com	TRUE	SCG1593750372
1	Matthews Peterson	mpeterson@gmail.com	TRUE	VCI8026324513

(4)

1.2 Champions for Change wants a list of the centres where the number of members per staff is greater than 30. Display the **CentreName**, **City** and number of members per staff. The number of members per staff can be calculated by dividing the number of members by the number of staff. Name the number of members per staff **MpS**. Round the **MpS** column to a whole number. Only display centres whose **MpS** is greater than 30. Sample output is given below:

Note only the first six records are shown.

CentreName	City	MpS
Foot Forward Three	Windhoek	31
Strive For Success	Serowe	36
Foot Forward Two	Windhoek	31
Place For Hope	Windhoek	33
Marching Forward	Henties Bay	35
Gaborone Winners One	Gaborone	31

(6)

1.3 Display the smallest and largest donations received in the year 2021. The correct output is shown below:

Minimum	Maximum
2563	99229

(4)

1.4 For tax purposes, Champions For Change needs a list of donors who have not received a tax certificate for any single donation and the amount they donated. They also need this information for any donation that was over 90000. Display the **DonorName**, **Amount** and **CertificateIssued** of all donations that have not had a certificate issued or are over 90000. Sample output is shown below:

Note only the first six records are shown.

DonorName	Amount	CertificateIssued
Matthews Peterson	19965	FALSE
Big Motors	32668	FALSE
Groceries and More	95615	TRUE
Gweneth Lebrun	54583	FALSE
Shania Julien	93254	TRUE
Getaway Travels	80132	FALSE

(6)

1.5 Dara Magee has made many donations and would like a record of cities where they have not made donations. Provide a list of cities where Dara Magee has not donated to any centres. You may hardcode Dara Magee's **DonorID** of **19** into the query. The correct output is shown below. Note, depending on the database used, your output may appear in a different order:

City
Gaborone
Gobabis
Henties Bay
Mariental
Maun
Omaruru
Selbi Phikwe
Serowe

(7)

1.6 Champions For Change wants to know who has frequently donated to specific centres. Display the **CentreName**, **DonorName** and number of times the donor has donated to that centre. Only display records where the donor has donated 3 or more times to a specific centre. The correct output is given below. Note, depending on the database used, your output may appear in a different order:

CentreName	DonorName	NumDonations
Flight of Champions	Matthews Peterson	3
On the Path	Getaway Travels	3

(8)

1.7 Donors whose email address is at yahoo.com have had their email address incorrectly stored in the **Contact** field. The domain was incorrectly entered as '@yaho.com'. Update the table to correct the email addresses so that they end in '@yahoo.com' (9)

1.8 Champions for Change has decided to add new adult centres in Windhoek. The centres will have the word 'Adult' prefixed to their name. For example, Place For Hope will become Adult Place For Hope. The number of staff working at the children's version of the centre will be the same as the adult version. Write a single query to add the new adult centres in Windhoek to the table called **tblCentres**.

The inserted records are shown below:

CentreID	CentreName	City	NumStaff	NumChildren
28	Adult Foot Forward Three	Windhoek	7	0
29	Adult Foot Forward Two	Windhoek	9	0
30	Adult Place For Hope	Windhoek	8	0
31	Adult Foot Forward One	Windhoek	6	0

(6)

50 marks

SECTION B OBJECT-ORIENTATED PROGRAMMING

Champions For Change runs a community centre for local underprivileged children. The children can become members of the centre at no cost. The centre has facilities for the members to play, exercise and learn safely. One daily cooked meal is provided to all members. Sometimes there is an excursion outside of the centre. On these days, all members will receive spending money instead of a cooked meal. The community centre provides money for some members while local businesses sponsor other members.

The centre-sponsored members all receive $40 in spending money. The businesses provide different amounts to each sponsored member. If a sponsored member receives less than $40 from the sponsor, the centre will give extra money to bring the total to $40. The amount of money will always be an integer.

For example:

1. Yakov Pemberton is not sponsored. They[1] receive $40 from the centre.
2. James Peterson is sponsored for $35. They receive $35 from their sponsor and $5 from the centre to bring their total to $40.
3. Stuart Redford is sponsored for $45. They receive $45 from the sponsor and nothing from the centre. This is more than the minimum, so the amount is not adjusted.

The dietary preferences of the children are represented by numbers as follows:

- **0** – standard – if the child has no specific preferences.
- **1** – halaal – if the child needs to eat halaal food.
- **2** – kosher – if the child needs to eat kosher food.
- **3** – vegetarian – if the child needs to eat vegetarian food.

If a child has a food preference that does not fit into the above categories, the food preference will be represented by any other number.

Member Class

This class will represent the basic details of children whose spending money is provided by the community centre.

- **memberName** – the full name of the member.
- **dob** – the date of birth of the member. This is stored in the format **yyyy-mm-dd**.
- **foodPref** – the number representing the member's food preference.
- **medAidName** – the name of the member's medical aid. A member who does not have medical aid will have a value **None** stored.

The details of the members are stored in a text file called **Members.txt.** A line of the file representing the data of a member has the format:

`<memberName>#<dob>#<foodPref>#<medAidName>`

[1] The gender neutral 'they' is used.

SponsoredMember Class

This class will represent the details of children whose spending money is provided by a sponsor. It is the subclass of the **Member** class. The **SponsoredMember** class has the following additional fields:

- **sponsorName** – the name of the business sponsoring the SponsoredMember.
- **spendingAmount** – the amount of money that the sponsor provides as spending money to the SponsoredMember.

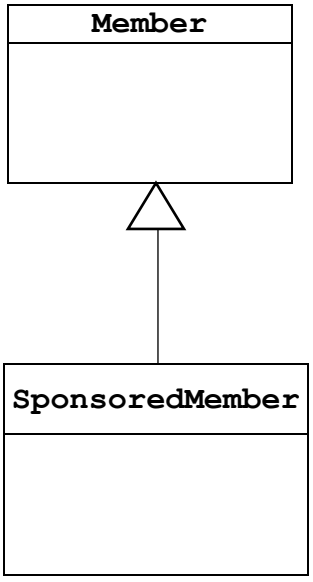
The details of the sponsored members are also stored in the **Members.txt** text file. A line of the file representing the data of a sponsored member has the following format:

<**memberName**>#<**dob**>#<**foodPref**>#<**medAidName**>#<**sponsorName**>#<**amount**>

The first ten lines of the text file, containing data about members and sponsored members are shown below. This data will be used to create **Member** and **SponsoredMember** objects in the questions that follow:

```
James Peterson#2006-10-16#2#Health Med#Wally's Furniture#35
Paulo Arrighetti#2009-12-19#2#None#Super Saver#30
Yakov Pemberton#2007-11-19#0#Sure Body
Valentyna Lagounov#2009-12-28#2#Health Med
Parth Baarsma#2012-07-07#0#None
Stuart Redford#2009-12-18#1#Protect Plus#Gifts and Cards#45
Alistair Fitzpatrick#2008-11-19#0#None
Anita Patel#2011-10-20#0#Health Med
Simon Smith#2010-09-09#0#None#Gifts and Cards#35
Mohammed Anand#2010-10-26#0#Sure Body#Super Saver#45
```

A UML diagram showing the relationship between the **Member** class and the **SponsoredMember** class is given below. The fields and methods of the **Member** class and the **SponsoredMember** class are given in the UML diagrams in Question 2 and 3.



QUESTION 2

Use the class diagram below to create a new class called **Member**. This class will be used to store the details of a member.

Member
Fields: - memberName : string - dob : Date - foodPref : integer - medAidName : string + FOOD_STANDARD = 0 : integer + FOOD_HALAAL = 1 : integer + FOOD_KOSHER = 2 : integer + FOOD_VEGETARIAN = 3 : integer
Methods: + Constructor(inMemberName : string, inDob : Date, inFoodPref : integer, inMedAidName : string) + getMemberName() : string + getFoodPref() : string + getAge() : integer + toString() : string

2.1 Create a new class named **Member** with the **memberName**, **dob**, **foodPref** and **medAidName** fields as shown in the class diagram. These fields must not be accessible outside the class. (3)

2.2 Add the constants **FOOD_STANDARD**, **FOOD_HALAAL**, **FOOD_KOSHER** and **FOOD_VEGETARIAN** as shown in the class diagram. (2)

2.3 Add a parameterised constructor method that will assign the values to the **memberName**, **dob**, **foodPref** and **medAidName** fields of the class. (4)

2.4 Add an accessor method for the **memberName** field of the class. (2)

2.5 Code a **getFoodPref** method to return a string representing the food preference of the **Member**. This method must return:

- '**standard**' if the food preference is **FOOD_STANDARD**
- '**halaal**' if the food preference is **FOOD_HALAAL**
- '**kosher**' if the food preference is **FOOD_KOSHER**
- '**vegetarian**' if the food preference is **FOOD_VEGETARIAN**
- '**other**' if the food preference is anything else (5)

2.6 Code a **getAge** method to return the current age of the **Member** in years. This method must return the correct current age of the **Member**. (4)

2.7 Code a **toString** method to return a string combining all the fields in the following format:

```
memberName<tab>age<space>"years old"<tab>"Food:"<space>
foodPreference<tab>"Medical Aid:"<space>medAidName
```

NB: The age must be given as an integer number of years. The food preference must be returned as a text value.

For example:

```
Yakov Pemberton    15 years old   Food: kosher   Medical Aid:
Sure Body
```

(4)
[24]

QUESTION 3

Use the class diagram below to create a new class called **SponsoredMember** that inherits from the **Member** class. This class will be used to store the details of a business-sponsored **Member**. The diagram below indicates the field and methods that are required.

SponsoredMember
Fields: - sponsorName : string - amount : integer
Methods: + Constructor(inMemberName : string, inDob : Date, inFoodPref : integer, inMedAidName : string, inSponsorName : string, inAmount : integer) + getSponsorName() : string + getAmount() : integer + toString() : string

3.1 Create a new class named **SponsoredMember** with the **sponsorName** and **amount** fields, as shown in the class diagram. **SponsoredMember** must inherit from the **Member** class. These fields must not be accessible outside the class. (4)

3.2 Write code to create a parameterised constructor method for the class that will accept parameters for the **memberName**, **dob**, **foodPref**, **medAidName**, **sponsorName** and **amount** fields. This method must use the superclass's constructor to instantiate the inherited fields and assign values to the **sponsorName** and **amount** fields. (5)

3.3 Create accessor methods for the **sponsorName** and **amount** fields. (2)

3.4 Create a **toString** method that will return a string that combines the values of the fields in the **SponsoredMember** class, including the inherited fields from the **Member** class. The method uses the **toString** method of the **Member** class. The format should be as follows:

```
memberName<tab>age<space>"years old"<tab>"Food:"<space>
foodPreference<tab>"Medical Aid:"<space>medAidName<tab>
"Sponsor Name:"<space>sponsorName<tab>"Amount:"<space>amount
```

For example:

```
James Peterson   17 years old    Food: kosher    Medical Aid:
Health Med    Sponsor Name: Wally's Furniture    Amount:
35
```

(3)
[14]

QUESTION 4

4.1 Create a class called **MemberManager**. (1)

4.2 Create two fields for the class. The first must be an array called **mArr** to store 100 **Member** and **SponsoredMember** objects. The second must be a counter called **size** to keep track of the number of objects added to the array called **mArr**. The fields should not be accessible outside the class. (4)

4.3 Create a constructor method that will read all the contents of the text file named **Members.txt** containing the information for members and sponsored members. Each line read from the file will add either one **Member** object or one **SponsoredMember** object to the array called **mArr**. The method should do the following:

- Check if the file **Members.txt** exists.
- Display an error message if the file does not exist.
- Open the file for reading.
- Loop through all the lines of the text file. In each iteration of the loop:
  - Read the line and split the data into separate parts.
  - Create either a **Member** object or a **SponsoredMember** object depending on the data on a line. Store the object in the next available position in the array called **mArr**.
- Update the counter variable called **size**. (13)

4.4 Write code to create a **toString** method. This method should return a string that contains details about all **Members** and **SponsoredMembers**; a blank line should separate each member's details. Use the **toString** methods created in the previous classes. (5)

4.5 Write code to create a method named **ageOfMember**. This method should accept the name of a member as a parameter. This method should return an integer representing the age in years of the member whose name was sent to the method.
If the member is not in the array, the method should return **-1**. (5)
[28]

QUESTION 5

5.1 Write code to create a text-based user interface called **MemberUI** that will allow simple input and output. (1)

5.2 Declare and instantiate a **MemberManager** object in an appropriate place in the code. (1)

5.3 Write code to call the appropriate method in the **MemberManager** class to display a list of all the **Member** and **SponsoredMember** objects. Sample output of the first five objects is below:

```
James Peterson     17 years old   Food: kosher   Medical Aid:
     Health Med    SponsorName: Wally's Furniture        Amount: 35
Paulo Arrighetti   13 years old   Food: kosher   Medical Aid:
     None          SponsorName: Super Saver      Amount: 30
Yakov Pemberton    15 years old   Food: standard   Medical
     Aid: Sure Body
Valentyna Lagounov 13 years old   Food: kosher   Medical Aid:
     Health Med
Parth Baarsma      11 years old   Food: standard   Medical
     Aid: None
```

(1)

5.4 Write code to call the appropriate method in the **MemberManager** class to display the age of **Stuart Redford**. The correct output is given below:

```
13
```

(3)
[6]

QUESTION 6

6.1 Code a constant called CENTRE_AMOUNT in the **MemberManager** class, which stores the default **Member** spending money amount of $40. (2)

6.2 Code a method called **totalMemberTypes** in the **MemberManager** class that returns the number of centre-sponsored, business-sponsored or combination-sponsored members. Each category below describes who provides the members spending money for an excursion:

- Entirely Centre
- Entirely Sponsored
- Sponsored and Centre

- A member with $40 spending money provided by the centre is in the **Entirely Centre** category.
- A sponsored member receiving either:
  - $40 or more are in the **Entirely Sponsored** category.
  - Less than $40 with additional centre-sponsored money to bring their total to $40 are in the **Sponsored and Centre** category.

Output should be in the following format:

```
"Entirely Centre:"<space>Entirely Centre<newline>
"Entirely Sponsored:"<space>Entirely Sponsored<newline>
"Sponsored and Centre:"<space>Sponsored and Centre<newline>
```

The correct output is shown below:

```
Entirely Centre: 11
Entirely Sponsored: 12
Sponsored and Centre: 7
```

(8)

6.3 In the **MemberUI** class, call the **totalMemberTypes** method in the appropriate place to display the string. (1)

[11]

QUESTION 7

An excursion to the Science Museum is being planned. A list of the sponsors who will provide spending money to one or multiple members is stored in the text file **Sponsors.txt**. Sponsors need to be billed for the total amount owed for the members they sponsor.

Sponsors.txt consists of 5 lines. Each line contains a sponsor name that matches the **sponsorName** data used when creating **SponsoredMember** objects.

7.1 Code a method called **spendingMoneyReport**.

- This method must accept one parameter, the filename of the list of sponsors.
- The method must create a string with the heading **"Spending Money Report"**.
- Underneath the heading, each sponsor must be listed. Calculate the total spending money amount a sponsor must provide to their sponsored members. Display this amount next to the sponsor's name.
- Remember that:
  - A member that is not sponsored will be provided with $40 by the centre.
  - The centre will provide the shortfall if a member is sponsored for less than $40.
  - If a member is sponsored $40 or more, the sponsor will only be billed $40.
- The code should use the constant from Question 6.1 to allow for efficient future changes to $40.

The correct output is shown below:

```
Spending Money Report
Wally's Furniture total = $110
Super Saver total = $230
Gifts and Cards total = $225
Deluxe Foods total = $115
Direct Motors total = $30
Centre Total = $490
```

(16)

7.2 Call the **spendingMoneyReport** method in the appropriate place and display the string. Send the filename **Sponsors.txt** as a parameter. (1)

[17]

100 marks

Total: 150 marks