NATIONAL SENIOR CERTIFICATE EXAMINATION
MAY 2023

INFORMATION TECHNOLOGY: PAPER I

Time: 3 hours 150 marks

PLEASE READ THE FOLLOWING INSTRUCTIONS CAREFULLY

1. This question paper consists of 15 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. Make sure 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, and 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 so that you can continue with the examination. If possible, try to explain the error to aid the marker.

8. When accessing files from within your code, DO NOT use full path names for the files, as this will create problems when the program is marked on a computer other than the one you are writing on. Merely refer to the files using their names and extensions, where necessary.

9. Your programs must be coded in such a way that they will work with any data and not just the sample data supplied or any data extracts that appear in the question paper. You are advised to look at the supplied data files carefully.

10. 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.

11. All data structures must be defined and declared by you, the programmer. You may not use components provided within the interface to store and later retrieve data.

12. 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.

13. 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 the original version is accidentally modified by your solution.

14. 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.

15. Make sure that your examination number appears as a comment in every program that you code as well as on every page of hard copy that you hand in.

16. Print a code listing of all the programs/classes that 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 for the printing of your work.

17. You should be provided with 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 provided in MS Access, JavaDB and MySQL format. Ensure that you are able to open the files with the packages that you will use to code your solutions to this examination.

    Section A:
    HousesDB.mdb
    HousesDB_JavaDB.sql
    HousesDB_MySQL.sql
    SQLAnswerSheet.rtf
    SQLBrowser.exe

    Section B:
    awards.txt

SCENARIO

At a local school, students are divided into groups called houses, with each student belonging to one house. You have been contacted by a friend who is part of the house leadership. The houses are named 'Lion', 'Buffalo', 'Elephant', 'Rhino' and 'Leopard'. Each house has two leaders, one of whom is the house captain in charge of the house. Each house has many fundraisers organised by teachers with a predetermined goal to be raised. At the end of a day's event, one of the house leaders must deposit the money raised. A fundraiser can be repeated multiple times.

Your friend has provided you with a database recording the 2021 fundraising data named **HousesDB**.

SECTION A STRUCTURED QUERY LANGUAGE

QUESTION 1

This database **HousesDB** contains fundraiser information, the details of the house leaders and their deposits in separate tables.

tblHouseLeaders describes the house leader's details.

FIELDS	DATA TYPE	DESCRIPTION
LeaderID	INTEGER	A unique autonumber identification number for each captain
Firstname	TEXT	The leader's first name
Surname	TEXT	The leader's surname
Grade	INTEGER	The leader's grade
House	TEXT	The leader's house
HouseCaptain	BOOLEAN	If they are the house captain

All the records in tblHouseLeaders

LeaderID	Firstname	Surname	Grade	House	HouseCaptain
1	Terence	Joubert	11	Lion	No
2	Grace	Ross	12	Lion	Yes
3	Constance	Swartz	11	Buffalo	No
4	Sibusiso	Scholtz	12	Buffalo	Yes
5	Mark	Schoeman	12	Elephant	No
6	Ayanda	Robertson	12	Elephant	Yes
7	Pieter	Evans	12	Rhino	No
8	Gail	Ntombela	12	Rhino	Yes
9	Lloyd	De Klerk	11	Leopard	No
10	Lindiwe	Kleynhans	12	Leopard	Yes

tblFundraisers contains the details of the fundraisers hosted by the school.

FIELDS	DATA TYPE	DESCRIPTION
FundraiserID	INTEGER	A unique autonumber identification number for each fundraiser
Title	TEXT	The title of the fundraiser
GoalAmount	DOUBLE	The goal amount to be raised in Rands
Organiser	TEXT	The teacher in charge of organising the fundraiser

The first 10 records of tblFundraisers

FundraiserID	Title	GoalAmount	Organiser
1	Car Wash	9000.00	Ms Jooste
2	Fashion Show	5000.00	Ms Green
3	Market Day	2000.00	Ms Kumalo
4	Art Show	8500.00	Mr Liebenberg
5	Mystery Dinner	5000.00	Mr Thompson
6	Golf Tournament	7000.00	Ms de Kock
7	Civvies Day	2000.00	Ms Naicker
8	Candy Sale	1000.00	Ms van Dyk
9	Talent Show	7500.00	Mr Beukes
10	Pancake Breakfast	2000.00	Ms Mthembu

tblDeposits contains the details of the money deposited by the house leaders for the different fundraisers.

FIELDS	DATA TYPE	DESCRIPTION
FundraiserID	INTEGER	The fundraiserID of the deposit. Part of the composite key for this table. A foreign key to the tblFundraisers
Amount	DOUBLE	The amount in Rands of the deposit
DepositType	TEXT	The type of deposit (cash, eft or credit card)
DepositDate	DATETIME	The date and time of the deposit. Part of the composite key for this table.
LeaderID	INTEGER	The LeaderID of the house leader that responsible for the Deposit. Part of the composite key for this table. A foreign key to the tblHouseLeaders.

The first 10 records of tblDeposits

FundraiserID	Amount	DepositType	DepositDate	LeaderID
1	547.00	credit card	2021/01/16 08:00	10
1	831.00	credit card	2021/01/29 14:50	2
1	313.00	cash	2021/10/26 13:49	2
1	631.00	cash	2021/11/04 13:06	5
2	201.00	credit card	2021/02/07 14:32	1
2	182.00	cash	2021/05/12 14:27	1
2	777.00	credit card	2021/06/01 10:06	2
2	566.00	cash	2021/09/22 17:20	6
3	577.00	cash	2021/05/30 12:44	5
3	598.00	cash	2021/06/17 16:57	2

1.1 Display the details of the leaders who are house captains in alphabetical order of house name.

The correct output is shown below:

LeaderID	Firstname	Surname	Grade	House	HouseCaptain
4	Sibusiso	Scholtz	12	Buffalo	Yes
6	Ayanda	Robertson	12	Elephant	Yes
10	Lindiwe	Kleynhans	12	Leopard	Yes
2	Grace	Ross	12	Lion	Yes
8	Gail	Ntombela	12	Rhino	Yes

(4)

1.2 List the fundraisers with the words 'show' or 'sale' in their titles.

The correct output is shown below:

FundraiserID	Title	GoalAmount	Organiser
2	Fashion Show	5000.00	Ms Green
4	Art Show	8500.00	Mr Liebenberg
8	Candy Sale	1000.00	Ms van Dyk
9	Talent Show	7500.00	Mr Beukes
13	Bake Sale	2500.00	Mr Mabena

(5)

1.3 The organisers for the fundraisers would like to create a hastag for social media. It should start with a '#' symbol, followed by the first and last letters of the fundraiser title. Thereafter, add the two letters to the right of the middle letter of the organisers name.
For example, the first and last letter of the 'Car Wash' is 'Ch' with the organiser 'Ms Jooste' who has 9 characters (including the space) in her name. The middle letter of her name will be the integer value of 9 divided by 2 producing 4. The next two characters after the 4th letter are 'oo' which must be added to 'Ch' to produce the code.
Display the title of the funraiser and the new hastag. *The hastag may contain spaces.*

The first 10 rows are shown below:

Title	Hashtag
Car Wash	#Choo
Fashion Show	#Fwre
Market Day	#Myum
Art Show	#Awbe
Mystery Dinner	#Mrom
Golf Tournament	#Gt K
Civvies Day	#Cyic
Candy Sale	#Cen
Talent Show	#Tweu
Pancake Breakfast	#Pthe

(6)

1.4 The bank charges 3.5% for any deposit that is not cash. Determine the bank charges (3.5%) of the non-cash deposits rounded to 2 decimal places.

The correct output is shown below:

TotalNonCashDeposits	BankCharges
30347.00	1062.15

(5)

1.5 Determine the fundraisers with a below-average goal amount. Display the title and the goal amount.

The correct output is shown below:

Title	GoalAmount
Market Day	2000.00
Civvies Day	2000.00
Candy Sale	1000.00
Pancake Breakfast	2000.00
Book Fair	2500.00
Bake Sale	2500.00
Valentines Day	3500.00

(5)

1.6 The organisers for the bake sale (FundraiserID = **13**) would like to list the house leaders who have not donated towards the fundraiser. Display the details of leaders who have not made any deposits for the bake sale fundraiser.

The correct output is shown below:

LeaderID	Firstname	Surname	Grade	House	HouseCaptain
3	Constance	Swartz	11	Buffalo	No
5	Mark	Schoeman	12	Elephant	No
7	Pieter	Evans	12	Rhino	No
9	Lloyd	De Klerk	11	Leopard	No

(4)

1.7 Determine the total amount deposited for each fundraiser. Display the title of the fundraiser and the total amount raised in a field called **TotalAmount**. Only show the fundraisers that have raised more than 2500 in total.

The correct output is shown below:

Title	TotalRaised
Bake Sale	4757.00
Civvies Day	5865.00
Golf Tournament	2782.00
Help the Planet	4117.00
Market Day	2924.00
Pancake Breakfast	4536.00
Talent Show	2794.00
Valentines Day	4663.00

(9)

1.8 A generous donor has agreed to match all the amounts deposited for the Help the Planet (FundraiserID = **15**) fundraiser. Write a query to duplicate all the deposits for this fundraiser. The payment type should be 'eft', with the current date and the same **LeaderID**.

Below are the newly added records to the table:

FundraiserID	Amount	DepositType	DepositDate	LeaderID
15	241	eft	2022/05/18	9
15	96	eft	2022/05/18	1
15	492	eft	2022/05/18	1
15	993	eft	2022/05/18	4
15	565	eft	2022/05/18	10
15	318	eft	2022/05/18	1
15	627	eft	2022/05/18	5
15	785	eft	2022/05/18	4

(7)

1.9 Write a query to add '-pm' to the deposit type for all cash deposits made after 12:00 noon. Below is a list of the affected records.

FundraiserID	Amount	DepositType	DepositDate	LeaderID
10	388.00	cash-pm	2021/04/27 17:30:00	7
2	182.00	cash-pm	2021/05/12 14:27:00	1
3	577.00	cash-pm	2021/05/30 12:44:00	5
13	159.00	cash-pm	2021/06/13 16:04:00	1
3	598.00	cash-pm	2021/06/17 16:57:00	2
13	899.00	cash-pm	2021/06/17 16:50:00	1
15	785.00	cash-pm	2021/08/12 12:12:00	4
9	405.00	cash-pm	2021/08/14 14:10:00	10
2	566.00	cash-pm	2021/09/22 17:20:00	6
12	591.00	cash-pm	2021/10/08 17:29:00	1
6	240.00	cash-pm	2021/10/25 13:22:00	6
1	313.00	cash-pm	2021/10/26 13:49:00	2
1	631.00	cash-pm	2021/11/04 13:06:00	5
9	559.00	cash-pm	2021/11/10 16:49:00	2
13	615.00	cash-pm	2021/11/14 17:18:00	8

(5)

50 marks

SECTION B OBJECT ORIENTATED PROGRAMMING

SCENARIO:

The school asks you to create a program to list students who receive service awards at the awards ceremony.

A student is given a service award for their service to the school, such as working in the library or helping with audio-visual technology at school events. Special service awards are given to students who served in a project outside of the school for an extended period during their high school career.

Depending on the number of hours served in a year, the service award is categorised into Bronze, Silver or Gold, where the minimum number of hours is a factor of 10:

Minimum hours (inclusive)		
Bronze	**Silver**	**Gold**
10 to 19	**20** to 29	**30** or more

To receive a special service award, the student must achieve a Gold service award.

A complete list of students eligible for awards is stored in a text file called **awards.txt.**

Below are the first 10 lines of the text file:

```
Clive Fourie#School Service#18
Mpho Ally#Community Service#41#Wild Life Sanctuary#2019-12-02
Shannon Carstens#Community Service#30#Hope Orphanage#2020-05-01
Sello Sibiya#School Service#18
Jeffrey Simelane#School Service#20
Charlotte Roux#Community Service#31#Gift of the Givers#2021-08-02
Lesley Hoosen#School Service#15
Christian Bezuidenhout#Library Assistant#29
Gert Muller#Library Assistant#27
Mpho Claassen#Library Assistant#23
```

Each line in the text file represents a service or special award recipient.

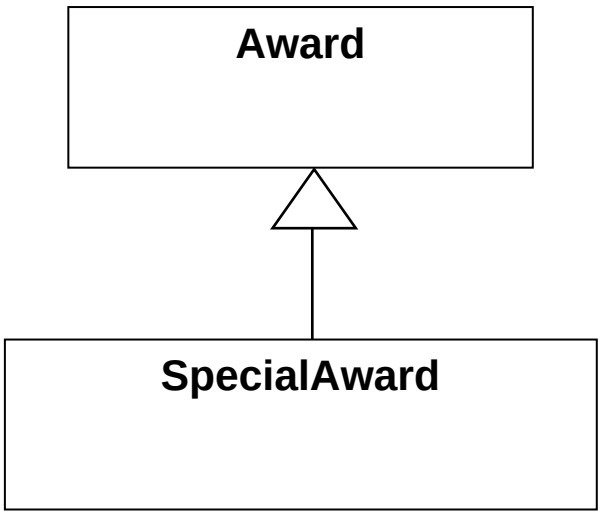
Award
The details for a service award recipient are stored in the text file as follows:
< fullname>#<descriptions>#<hourslogged>

- **Fullname:** a string storing the student's name and surname receiving the service award.
- **Description:** a string describing the project or school service of the student.
- **HoursLogged:** an integer storing the annual total hours volunteered by the student.

SpecialAward
The details of a special service award recipient are stored in the text file as follows:
< fullname>#<descriptions>#<hourslogged>#<project>#<datestarted>

- **Fullname:** a string storing the student's name and surname receiving the service award.
- **Description:** a string describing the project or school service of the student.
- **HoursLogged:** an integer storing the annual total hours volunteered by the student.
- **Project:** a string storing the external project name.
- **DateStarted:** the date the student first started with the project in the format yyyy-MM-dd.

You will need to create two classes with the parent **Award** class and the child **SpecialAward** class as shown in the diagram below:



QUESTION 2

Use the class diagram below to create a class called **Award**. This class will be used to create objects to store the details of an award recipient. The diagram indicates the required fields and methods.

Award
Fields: – fullname : string – description : string – hoursLogged : integer – minHourFactor : integer = 10
Methods: + Constructor(inFN : string, inDN : string , inHD : integer) + getFullName() : string + getDescription() : string + getHoursLogged() : integer + getMinHourFactor() : integer + setMinHourFactor(inMHF : integer) + getAwardColour() : string + toString() : string

2.1 Create a new class called **Award** with **fullname**, **description** and **hoursLogged** fields as indicated above. (4)

2.2 Add a **static/class** field called **minHourFactor** as shown in the diagram. (1)

2.3 Code a constructor method for the class that will accept the string **inFN** for the **fullname** field, a string named **inDN** representing the **description** field and an integer named **inHD** representing the total **number of hours logged**. Using these parameters assign all the values for the fields of the class. (3)

2.4 Create accessor/getter methods for the **fullname**, **description** and **hoursLogged** fields of the class. (3)

2.5 Code an accessor and mutator method for the static field **minHourFactor**. (2)

2.6 Create the method **getAwardColour** that will return a string containing the type of award colour (Gold, Silver, Bronze) that the student has earned using the **minHourFactor** field. The method must return 'Gold' if they have 30 or more hours, 'Silver' if they have 20 or more hours and 'Bronze' for 10 or more hours. (5)

2.7 Add a **toString** method to the class that will return a **string** containing all the information of the class in the following format:

fullname description hoursLogged" hrs (**"award colour"**)**"**

For example
Clive Fourie Community Service 12 hrs (Silver) (4)

[22]

QUESTION 3

Use the class diagram below to create a class called **SpecialAward**. This class will inherit from the **Award** class and will be used to create objects that will store the details of a special award recipient. The diagram below indicates required fields and methods.

SpecialAward
Fields: – project : string – dateStarted : Date + numSpecialAwards : integer = 0
Methods: + Constructor(inFN:string,inDN:string,inHD:integer,inPT:string, inDS : Date) + getProject() : string + getDateStarted() : Date + getNumYears() : integer + toString() : string

3.1 Create a new class called **SpecialAward** that extends the **Award** class. (2)

3.2 Add the **project** and **dateStarted** fields as indicated by the diagram. Note that an appropriate date object type must be used to store the date information. (3)

3.3 Add a **static/class** field called **numSpecialAwards** as shown in the diagram. (1)

3.4 Code a constructor method that will initialise the fields **fullname**, **description** and **hoursLogged** of the **Award** parent class and the fields **project** and **dateStarted** of the **SpecialAward** child class. Increment the static field **numSpecialAwards**. (5)

3.5 Create an **accessor** method for the **project** and **dateStarted** fields. (2)

3.6 Code a method called **getNumYears** to determine the number of years between the **dateStarted** field and **18 May 2023**. (5)

3.7 Code a **toString** method that will override the parent class's **toString** method by appending the project and number of years to the parent's toString and return a string in the following format:

fullname description hoursLogged" hrs (**"award colour"**)"
project "project" **number of years** "Years"

Example

Mpho Ally Community Service 21 hrs (Silver) Wild Life Sanctuary project 3 Years (4)

[22]

QUESTION 4

4.1 Create a new class called **AwardManager**. (1)

4.2 Add two instance variables to this class:
- An **array** capable of storing 50 **Award** or **SpecialAward** objects. Name this array **awArr**.
- An integer called **size** that will keep track of the number of objects added to the array. These two variables should not be accessible outside the class. (4)

4.3 Code a constructor method for the class that will read the contents of the file **'awards.txt'**. Each line of the file contains information on either a **Award** or a **SpecialAward**. Your method should do the following:

- Read each line from the file and extract the data.
- Instantiate the appropriate type of object (**Award** or a **SpecialAward**) using the data extracted from the line
- Add the **Award** or **SpecialAward** object to the next available position in the array.
- Increase the **size** appropriately. (11)

4.4 Code a **toString** method to combine the values of each object in the array **awArr** into a string. Each award or special award's information should appear on a separate line. Use the object's **toString** methods that you created in Questions 2.7 and 3.7.

Sample output is shown in Question 5.3. (5)

4.5 Create a method named **sort** to sort all the objects in the array called **awArr** in alphabetical order of their full names. (7)

4.6 Code a method called **deleteAward** to remove an element from the array given the element's integer position in the array. (5)

[33]

QUESTION 5

5.1 Create a class called **AwardsUI** that will have a simple text-based user interface and output. (1)

5.2 Create a **AwardsManager** object in an appropriate place in the class using an appropriate method. (1)

5.3 **Sort** and display all the awards in alphabetical order by their full names of the students. (2)

```
Angelique Brand Feeding Scheme 21 hrs (Silver)
Anthony Zwane AV Crew 17 hrs (Bronze)
Boitumelo Thomas Feeding Scheme 28 hrs (Silver)
Bongani Moloi School Service 27 hrs (Silver)
Brent Cele Leadership 30 hrs (Gold) Mentorship Program project 3 Years
...
Sello Sibiya School Service 18 hrs (Bronze)
Shannon Carstens Community Service 30 hrs (Gold) Hope Orphanage project 3 Years
Thembi Nelson Community Service 22 hrs (Silver) Big Cat Sanctuary project 4 Years
Tracey Stevens AV Crew 20 hrs (Silver)
Yolandi Moloto School Service 15 hrs (Bronze)
```

5.4 In the **AwardsUI** class display the number of special service awards. Use the static/class fields you created in Questions 3.3.

Your output should appear as:

```
Number of special award Recipients: 11
```

(1)

[5]

QUESTION 6

6.1 The school is considering changing the criteria for special awards by increasing the minimum number of hours for 10 to 12, so that bronze will start at 12, silver at 24 and gold at 36. Code a method called **changeRequirements** to determine which students will be affected by the proposed criteria. The method will return a list of students who do not achieve special awards (they no longer achieve a gold service award) and a list of students who will receive no service award. In this method perform the following:

- Change the static field **numMinHours** to 12 using the appropriate method in the Award class.
- Remove any student who has less than **numMinHours**.
- Convert any student who previously qualified for a special service award to a service award if they do not achieve a gold category service award based on the new minimum hours.
- Return a string with a list of student names who were deleted and a separate list of students who no longer receive special service awards and have been converted to service awards. (16)

6.2 In the **AwardsUI** class call the method **changeRequirements** to display the lists of those who were deleted and those who no longer achieve special service awards.

Your output should appear as follows:

```
Students who do not achieve a Special Award
Brent Cele
Charlotte Roux
Kelly Delport
Kyle Ndebele
Liezel Marais
Lydia Harmse
Shannon Carstens
Thembi Nelson

Deleted students
Carla Hattingh
Mike Dhlamini
```

(2)

[18]

100 marks

Total: 150 marks