basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NATIONAL SENIOR CERTIFICATE

GRADE 12

INFORMATION TECHNOLOGY P1

NOVEMBER 2013

MARKS: 120

TIME: 3 hours

This question paper consists of 16 pages and 3 annexures.

INSTRUCTIONS AND INFORMATION

1. The duration of this examination is three hours. Because of the nature of this examination it is important to note that you will not be permitted to leave the examination room before the end of the examination session.

2. No distinction has been made between the two programming languages in this question paper with regard to the formulation of the questions on programming. Where required, specific instructions have been provided for Delphi and Java candidates respectively.

3. You require the files listed below in order to answer the questions. The invigilator/teacher will tell you where to find them.

Question1_Delphi:
Question1DB.mdb
Question1P.dpr
Question1P.res
Question1U.dfm
Question1U.pas
tblDanceCouples.txt
tblResults.txt

Question1_Java:
Question1.java
Question1DB.mdb
tblDanceCouples.txt
tblResults.txt
TestQuestion1.java

Question2_Delphi:
Question2P.dpr
Question2P.res
Question2U.dfm
Question2U.pas
uDanceCouple.pas

Question2_Java:
DanceCouple.java
TestQuestion2.java

Question3_Delphi:
DataQ3.txt

Question3_Java:
DataQ3.txt
MenuQ3_Java.txt

If you received the files above on an external medium (CD, stiffy or flash drive) write your examination number on the label.

4. Type your examination number as a comment in the first line of each program file that contains your programming code.

5. Your program should always be coded to answer the question in such a way that it will run with different sets of input data.

6. Read ALL the questions carefully. Do not do more than the questions require.

7. Read the entire question before you answer any subquestions.

8. Save your work at regular intervals as a precaution against power failures.

9. During the examination, you may use the manuals originally supplied with the hardware and software. You may also use the HELP functions of the software. Java candidates may use the Java API files. You may NOT use any other resource material.

10. At the end of this examination session, you must hand in the external medium with all your work saved on it OR you must make sure that all your work has been saved on the network as explained to you by the invigilator/teacher.

11. Ensure that all files saved on the external medium or network can be read.

12. If required, make printouts of the programming code for all of the questions you have done.

13. All printing of the questions that you have done will take place within an hour of the completion of this examination.

SCENARIO

Your community has planned different events to raise funds for the local community centre. One event planned is an annual dance competition.

QUESTION 1: PROGRAMMING AND DATABASE

The dance competition is held over a period of 12 weeks. It starts with 14 competing dance couples. Some of the competitors are professional dancers. Each week one of the dance couples is eliminated based on their performance at the weekly elimination round. During the final week of the competition two elimination rounds will take place to determine who the winning dance couple will be.

A Microsoft Office Access database named **Question1DB.mdb**, two text files (**tblDanceCouples.txt** and **tblResults.txt**) and an incomplete program are given in the folder named **Question1_XXXX**, where XXXX refers to the programming language you have studied.

The design of the tables in the **Question1DB** database and sample data from each table are given in **ANNEXURE A**.

Do the following:

- Make a backup copy of the **Question1DB** database BEFORE you start answering the questions. You will need a copy of the original database to be able to test your program thoroughly.
- Rename the folder for QUESTION 1 by replacing the name of the programming language you have studied with your examination number.
- Open the incomplete program for QUESTION 1.
- Enter your examination number as a comment in the first line of the program file.
- Compile and execute the program. The interface displays eight menu options: Option A to Option G and a 'Quit' option.

  NOTE:

  - An error message will be displayed if any of Option A to Option G are selected because of the incomplete SQL statements.
  - If you experience any problems using the database or connecting to the database, refer to **ANNEXURE B (Delphi)/ANNEXURE C (Java)** for troubleshooting hints.
  - If you still experience database problems, you must nevertheless do the SQL code and submit it for marking. **Marks will only be awarded for the programming code that contains the SQL statements.**

- Complete the code for each menu option by formulating an appropriate SQL statement to display the respective query results as described in QUESTIONS 1.1 to 1.7 below.

  NOTE: The code to some input statements and the code to execute the SQL statements and display the results of the queries have already been written as part of the given code.

1.1 **Menu Option A**

Display all the information in the **tblResults** table. Sort the data alphabetically according to the **TypeOfDance** field and then according to the **RoutineNo** field in descending order.

Example of the output of the first four records:

RoutineNo	Week	Round	DanceCoupleID	TypeOfDance	Song	Score	Result
116	12	2	8	American Smooth	Can't Help Falling in Love	39	Final
107	11	1	11	American Smooth	Time After Time	36	Safe
103	10	1	14	American Smooth	Singin' in the Rain	37	Safe
101	10	1	10	American Smooth	Pretty Woman	34	Safe

:

(3)

1.2 **Menu Option B**

Determine the dance routines with scores between 25 and 35 (including both) that were performed during the fifth week and the ninth week of the competition. Display the **RoutineNo**, **Week, TypeOfDance** and the **Score** for these routines.

Example of the output of the first four records:

RoutineNo	Week	TypeOfDance	Score
55	5	Tango	32
57	5	Paso Doble	31
58	5	American Smooth	35
60	5	Tango	34

:

(4)

1.3 **Menu Option C**

Allow the user to enter the name of a type of dance, for example Rumba.

Count and display the number of times this type of dance was performed during the competition. Name the calculated field **NumberOfPerformances**. Display the **TypeOfDance** and the **NumberOfPerformances**.

Example of the output if Rumba were entered by the user:

TypeOfDance	NumberOfPerformances
Rumba	7

(5)

1.4 **Menu Option D**

Determine the titles of the songs starting with the word 'Love' or containing the word 'you' or variations of the word 'you' that professional dance couples performed their dance routines to. Display the titles of the songs and the names of the dance couples who performed routines to these songs.

Example of the output:

Song	DancePartner1	DancePartner2
Love Man	Robbie	Ola
Love Ain't Here Anymore	Robbie	Ola
Let Me Entertain You	Robbie	Ola
You Sexy Thing	Robbie	Ola
You Can't Stop the Beat	Anita	Robin
I've Got You Under My Skin	Anita	Robin
Love Potion No. 9	Chelsee	Pasha
Spice Up Your Life	Chelsee	Pasha
Because of You	Chelsee	Pasha

(7)

1.5 **Menu Option E**

Calculate and display the average score for each dance couple in a calculated field named **AverageScore**.

The average score for each dance couple must be calculated using the total score and the number of dance routines they performed. Display the average score, correct to three decimal places. Display the **DanceCoupleID** and the **AverageScore** of all the dance couples.

Example of the output of the first four records:

DanceCoupleID	AverageScore
1	32.727
2	23.000
3	24.833
4	22.714
:	

(6)

1.6 **Menu Option F**

Display a list with the names of all the dance couples who were eliminated up to the final week (week 12) of the competition. The names of each dance couple must appear only once on the list.

Example of the output (on the next page):

DancePartner1	DancePartner2
Alex	James
Anita	Robin
Audley	Natalie
Dan	Katya
Edwina	Vincent
Holly	Artem
Lulu	Brendan
Nancy	Anton
Robbie	Ola
Rory	Erin
Russell	Flavia

(5)

1.7 **Menu Option G**

Dance Couple 8 are announced as the winners of the competition. Update the content of the **Result** field for Dance Couple 8 to contain the word 'WINNERS' for the second round of the final week of the competition.

Once the records have been updated successfully, a suitable message will be displayed. The code for this message is supplied. (5)

NOTE: If you want to test the menu options at this stage, use your backup copy of the **Question1DB** database.

- Enter your examination number as a comment in the first line of the file containing the SQL statements.
- Save your program.
- Make a printout of the code, if required.

[35]

QUESTION 2: OBJECT-ORIENTED PROGRAMMING

A program is needed to provide information about how the judges score the dance couples.

The files required for this question can be found in the folder named **Question2_XXXX**, where XXXX refers to the programming language you have studied.

You have been provided with an incomplete program that consists of:

Delphi	Java
• A class unit named **uDanceCouple** which describes the attributes of a dance couple and contains some methods	• An object class named **DanceCouple** which describes the attributes of a dance couple and contains some methods
• A main form unit named **Question2U**	• A test class named **TestQuestion2**

The given **uDanceCouple/DanceCouple** class contains the declaration and coding of:

- Three attributes describing a dance couple
- One constructor
- Three get/accessor methods

The three attributes describing a dance couple are:

Description	Names of attributes	
	Delphi	Java
The name of the first dance partner	fDanceP1	dance_p1
The name of the second dance partner	fDanceP2	dance_p2
The character A, B, C or D indicating the professional dance status of the couple	fProfessional	professional

NOTE: The professional dance status can be one of the following:

- A – Both dance partners are professional dancers.
- B – The first dance partner is a professional dancer.
- C – The second dance partner is a professional dancer.
- D – Neither of the dance partners is a professional dancer.

Any other character besides A, B, C or D will also be accepted and therefore the characters should not to be validated.

Do the following:

- Rename the folder for QUESTION 2 by replacing the name of the programming language you have studied with your examination number.

Delphi	Java
• Open the incomplete project file **Question2P.dpr**.	• Open the **TestQuestion2.java** test class file.
• Open the **uDanceCouple.pas** class unit.	• Open the incomplete object class **DanceCouple.java**.
• Add your examination number as a comment in the first line of both the **uDanceCouple.pas** class unit and the **Question2U.pas** main form unit.	• Add your examination number as a comment in the first line of both the **DanceCouple.java** object class and the **TestQuestion2.java** test class.

- Compile and execute the program. The interface displays four menu options: Option A, Option B, Option C and a 'Quit' option.

NOTE: Run the menu options in sequence when you test the program, in other words, first Option A, then Option B and then Option C.

2.1 Do the following to complete the code in the given **uDanceCouple/ DanceCouple** class:

2.1.1 Write code for a **constructor** using parameters to initialise the three attributes of the class. The attributes are the names of the two dance partners and the professional dance status of the dance couple (see previous page). (5)

2.1.2 Write code for a method named **getWeighting** that will determine and return a weighting value for a dance couple based on the professional dance status of the dance couple. A higher weighting value is to the benefit of the dance couple. The following applies:

- If neither dance partner is a professional dancer, the weighting is set to 3.
- If one of the dance partners is a professional dancer, the weighting is set to 2.
- If both dance partners are professional dancers, the weighting is set to 1.
- If the professional dance status is any character other than A, B, C or D, the weighting is set to 0. (8)

2.1.3 Write code for a method named **calcFinalScore** that will receive an array that contains the scores of the four judges for a dance performance as a parameter. This method must calculate and return the final score for the dance routine.

The final score for a dance routine is calculated as follows:

- The scores of the **first** judge and the **third** judge are each multiplied by the weighting value of the dance couple, which is determined by their professional dance status.
- The scores of all four judges are then added up once the weighting value has been taken into account. (7)

2.1.4 Write code for a **toString** method that will construct and return a string with labels and information about a dance couple object in the following format:

```
Dance couple: <dance partner 1> & <dance partner 2>
Professional dance status: <character indicating the professional dance status of the dance couple>
```

Example of the output for Sarah and John with C as their professional dance status:

```
Dance couple: Sarah & John
Professional dance status: C
```

(6)

2.2 Do the following to complete the code in the main form unit (Delphi)/test class (Java):

2.2.1 Write code to complete **Menu Option A** to do the following:

- Allow the user to enter the names of the dance couple – first the name of dance partner 1 and then the name of dance partner 2.

- Allow the user to enter the professional dance status (A, B, C or D) of the dance couple.

  The professional dance status can be one of the following:

  - A – Both dance partners are professional dancers.
  - B – The first dance partner is a professional dancer.
  - C – The second dance partner is a professional dancer.
  - D – Neither of the dance partners is a professional dancer.

  NOTE: Any other character can be accepted.

- Declare and instantiate a dance couple object using the user input values.

- Display the information for a dance couple as shown in the example below.

Example of the output if the names of the dance partners are Sarah and John and their professional dance status is C:

```
Dance couple: Sarah & John
Professional dance status: C
```

(8)

2.2.2 Write code to complete **Menu Option B** to do the following:

- Declare an array of integers to store the individual scores of the four judges for a dance routine. The scores have a possible maximum value of 10.

  Example of the four possible scores: 8, 6, 7, 9

- Allow the user to enter the four scores one by one. Assign a value of zero to a score if an invalid score, for example a negative score or a score greater than 10, is entered.

- Call the relevant methods to calculate the final score for the dance routine and display the output in the following format:

```
Dance couple: <dance partner1> & <dance partner2>
Scores from the judges: <score1> <score2> <score3> <score4>
Weighting value: <weighting value>
Final score: <calculated final score>
```

Example of the output for the dance couple Sarah and John if the scores from the judges are 8, 6, 7 and 9:

```
Dance couple: Sarah & John
Scores from the judges: 8 6 7 9
Weighting value: 2
Final score: 45
```

(10)

2.2.3 Write code to complete **Menu Option C** to do the following:

Write the names of the dance couple and their final score to a new text file named **Score.txt** in the following format:

<dance partner 1>,<dance partner 2>,<final score>

Example of data in **Score.txt**:

```
Sarah,John,45
```

Display a message indicating that the information has been written to a text file. (7)

- Make sure that your examination number is entered as a comment in the first line of the class unit (Delphi)/object class (Java) as well as the main form unit (Delphi)/test class (Java).
- Save all the files.
- Make a printout of the code, if required. Print the class unit (Delphi)/object class (Java) and the main form unit (Delphi)/test class (Java). **[51]**

QUESTION 3: PROBLEM-SOLVING PROGRAMMING

During the fifth week of the dance competition the organisers decide to allow members of the public to vote for their favourite dance couple. Each dance couple has a unique competition number. People who are 18 years or older can vote by sending an SMS with the number of their favourite dance couple. Four of the original 14 couples were already eliminated during previous rounds. The public can only vote for the dance couples who are still in the competition. The dance couple with the lowest number of votes from the public may be eliminated during this round.

The folder for QUESTION 3 contains a text file named **DataQ3.txt** for Delphi and two text files (**DataQ3.txt** and **MenuQ3.txt**) for Java.

Each line of text in the text file (**DataQ3.txt**) contains information of an SMS that was received in the following format:

<name of the person who voted>;<the person's age in years>#<cellphone number>;<the competition number of the dance couple the person voted for>

Example of the data for the first three SMSs in the text file named **DataQ3.txt**:

```
Armando Harris;56#+27848283872;8
Luca Huonti;49#+2585514394;1
Venere Laconi;47#+27768984873;12
```

The data of the first two lines of text can be interpreted as follows:

- Armando Harris, aged 56, used cellphone number +2784 828 3872 to vote for Dance Couple 8.
- Luca Huonti, aged 49, used cellphone number +258 551 4394 to vote for Dance Couple 1.

Do the following:

- Rename the folder for QUESTION 3 by replacing the name of the programming language you have studied with your examination number.
- Create a new program/project/application.
- Enter your examination number as a comment in the first line of the program file(s) you have created that will contain your code.
- Save the program file(s) using the question number as part of the filename in the renamed folder for QUESTION 3.

- Develop an interface as follows:

Delphi	Java
• Use a *MainMenu* component and develop an interface with three menu options: Option A, Option B and a 'Quit' option.	• Copy the given code for displaying a menu from the **MenuQ3.txt** text file. The interface needs to display three menu options: Option A, Option B and a Quit option.
• Add a *RichEdit* component called **redQ3** on the form to display information. Change the *Align* property to *alClient*.	
• Use the following code to complete the 'Quit' menu option: `Application.Terminate;`	• Use the following code to complete the 'Quit'-menu option: `System.exit(0);`

- Complete the code for each menu option as follows:

NOTE: Run the menu options in sequence when you test the program, in other words, first Option A and then Option B.

3.1 **Menu Option A**

Some of the SMSs captured in the text file are invalid. An SMS is invalid when:

- The vote was cast for a dance couple who have already been eliminated
- The vote was cast by a person who is younger than 18 years

Allow the user to enter the competition numbers of the four dance couples already eliminated from the competition. The user can enter any four values in the range from 1 to 14. Validate the values entered to comply with the following conditions:

- The value must be a number and not a character.
- The value must be inside the range of 1 to 14.

NOTE:
- The program may only continue once all four input values have been validated.
- The validated four numbers need to be available to be used in Menu Option B.

If the cellphone number starts with the prefix '+27', the vote is regarded as a vote from within the borders of South Africa. Otherwise it is regarded as an international vote.

Display a numbered list with the contact details of all persons who cast valid votes. Indicate if the person cast an international vote by displaying the phrase 'International vote' as part of his/her contact details as shown in the example below.

Display the total number of valid and invalid votes at the bottom of the list.

Example 1: An example of the output after Option A has been executed and Dance Couples 1, 2, 3 and 4 have been eliminated:

```
No.   Name               Cellphone number
1     Armando Harris     +27848283872
2     Venere Laconi      +27768984873
3     Gabriella Thao     +27839161414
:
302   Lucas Herder       +2581101571     International vote
303   Giacobbe Spina     +2582751425     International vote
304   Rachel Delarosa    +27833054860

Invalid votes: 196
Valid votes: 304
```

Example 2: An example of the output after Option A has been executed and Dance Couples 1, 5, 6 and 8 have been eliminated:

```
No.   Name               Cellphone number
1     Venere Laconi      +27768984873
2     Gabriella Thao     +27839161414
3     Tim Anger          +2645519196     International vote
:
272   Estotz Lizarazu    +27833247494
273   Lucas Herder       +2581101571     International vote
274   Rachel Delarosa    +27833054860

Invalid votes: 226
Valid votes: 274
```

(22)

3.2 **Menu Option B**

The organisers need a summarised report that lists the total number of valid votes that were cast for each dance couple. More than one dance couple may have the same number of votes. The dance couple(s) with the lowest number of valid votes may be eliminated.

Indicate the following in the report:

- A list with headings and subheadings with the total number of votes that were cast for each dance couple. Indicate the four dance couples who have already been eliminated.
- The dance couple(s) who may be eliminated next based on the least number of votes.

Write code to generate and display the required report as indicated in the following examples.

Example 1: An example of the output after Option A has been executed and Dance Couples 1, 2, 3 and 4 have been eliminated:

```
Votes received during week 5:
Dance couple     Votes
1                (Eliminated)
2                (Eliminated)
3                (Eliminated)
4                (Eliminated)
5                30
6                30
7                30
8                43
9                27
10               25
11               25
12               33
13               25
14               36

Dance couple(s) who may be eliminated next: 10 11 13
```

Example 2: An example of the output after Option A has been executed and Dance Couples 1, 5, 6 and 8 have been eliminated:

```
Votes received during week 5:
Dance couple     Votes
1                (Eliminated)
2                31
3                22
4                20
5                (Eliminated)
6                (Eliminated)
7                30
8                (Eliminated)
9                27
10               25
11               25
12               33
13               25
14               36

Dance couple(s) who may be eliminated next: 4
```

(12)

- Enter your examination number as a comment in the first line of the program.
- Save all the files.
- Make a printout of the code, if required.

[34]

TOTAL: 120

ANNEXURE A: DATABASE STRUCTURE AND SAMPLE DATA

This annexure shows the database structure and sample data for the tables contained in the **Question1DB.mdb** database used in **QUESTION 1**.

tblResults: This table contains data on the results of all the dance routines that were performed during the twelve weeks of the competition.

Field Name	Type	Size	Description
RoutineNo	Autonumber	Long Integer	A unique number for each dance routine that was performed during the competition
Week	Number	Integer	A number indicating the week during which the dance routine was performed. The competition runs over 12 weeks.
Round	Number	Integer	A number indicating the round during which the routine was performed. The final week has two rounds. All the other weeks have only one round.
DanceCoupleID	Number	Long Integer	A unique number to identify the dance couple who performed the dance item
TypeOfDance	Text	30	Type of dance that was performed, for example waltz, samba, cha-cha, et cetera
Song	Text	40	The title of the song used during the dance routine
Score	Number	Integer	The total score from the judges for the dance routine
Result	Text	15	Indicates the result after the dance routine was performed. The result can be one of the following: • Safe – The dance couple are still in the competition. • Eliminated – The dance couple have been eliminated. • Final – The dance couple are in the final week.

Example data:

RoutineNo	Week	Round	DanceCoupleID	TypeOfDance	Song	Score	Result
1	1	1	1	Cha-cha	Who's That Chick?	28	Safe
2	1	1	2	Waltz	Are You Lonesome Tonight?	24	Safe
3	1	1	3	Cha-cha	I've Got The Music in Me	17	Safe
4	1	1	4	Waltz	Angel	20	Safe
5	1	1	5	Cha-cha	Bad Boys	19	Safe
6	1	1	6	Waltz	Three Times a Lady	28	Safe
7	1	1	7	Cha-cha	Venus	21	Safe
8	1	1	8	Cha-cha	Moves Like Jagger	28	Safe

:

tblDanceCouples: This table contains the data of the 14 dance couples who took part in the dance competition.

Field Name	Type	Size	Description
DanceCoupleID	AutoNumber	Long Integer	A unique number assigned to each dance couple
DancePartner1	Text	10	The first name of dance partner 1
DancePartner2	Text	10	The first name of dance partner 2
ProfessionalDancers	Text	1	The symbol that indicates whether the dance partners are professional dancers. The following applies: • B – Both dance partners are professional dancers. • O – One of the dance partners is a professional dancer. • N – Neither dance partner is a professional dancer.
PreviousDancePartners	Yes/No		An indication whether the dance couple were dance partners in previous dance competitions

Example data:

DanceCoupleID	DancePartner1	DancePartner2	ProfessionalDancers	PreviousDancePartners
1	Holly	Artem	B	☑
2	Dan	Katya	O	☑
3	Lulu	Brendan	N	☑
4	Audley	Natalie	O	☐
5	Robbie	Ola	B	☑
6	Anita	Robin	B	☑
7	Russell	Flavia	N	☐
8	Harry	Aliona	N	☐
9	Rory	Erin	N	☑
10	Alex	James	O	☑
11	Chelsee	Pasha	B	☐
12	Edwina	Vincent	O	☐
13	Nancy	Anton	B	☐
14	Jason	Kristina	N	☐

ANNEXURE B: DELPHI – TROUBLESHOOTING DATABASE PROBLEMS

B.1 If you cannot use the database provided:

- Create your own database with the name **Question1DB** that includes two tables, named **tblResults** and **tblDanceCouples** in the same folder as your program for QUESTION 1.
- Import the two text files (**tblResults.txt** and **tblDanceCouples.txt**) to use as data for the different tables.
- The first line in the text files contains the field names to be used.

B.2 If your program cannot establish connectivity with the database, make sure that the database **Question1DB** is in the same folder as your program for QUESTION 1. If this is not the case, copy the database file into the same folder as your program.

B.3 If your program establishes connectivity with the database but no data is displayed:

- Click on the ADOQuery component named **qryRec**.
- Click on the ellipsis button (three dots) to the right of the 'ConnectionString' property in the Object Inspector.
- Click on the 'Build' button, which takes you to the Data Link Properties dialogue window.
- Click on the 'Provider' tab to open the 'Provider' tab sheet and select 'Microsoft Jet 4.0 OLE DB Provider'. Click on the 'Next' button.
- The 'Connection' tab sheet will be displayed. The first option on the 'Connection' tab sheet provides an ellipsis button (three dots) that allows you to browse and look for the **Question1DB** file. You will find this file in the **Question1** folder. Once you have found it, select the **Question1DB** file and then click on the 'Open' button.
- Remove the username 'Admin'.
- Click on the 'Test Connection' button.
- Click 'OK' on each of the open dialogue windows.

ANNEXURE C: JAVA – TROUBLESHOOTING DATABASE PROBLEMS

C.1 If you cannot use the given database:

- Create your own database with the name **Question1DB** that includes two tables, named **tblResults** and **tblCouples** in the same folder as your program for QUESTION 1.
- Import the two text files (**tblResults.txt** and **tblCouples.txt**) to use as data for the different tables.
- The first line in the text files contains the field names to be used.

C.2 If your program cannot establish connectivity with the database make sure that the database **Question1DB** is in the same folder as your program for QUESTION 1. If this is not the case, copy the database file into the same folder as your program.

C.3 If you cannot establish connectivity with the database with the given files, use the following source code to ensure database connectivity:

```
try
  {
     Class.forName ('sun.jdbc.odbc.JdbcOdbcDriver');
     String filename = 'Question1DB.mdb';
     String database = 'jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=';
            database += filename.trim () + ';DriverID=22;READONLY=true}';
     Connection conn = DriverManager.getConnection (database, '', '');
  }
  catch (Exception e)
  {
     System.out.println ('Unable to connect with the database');
  }
```