basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NATIONAL SENIOR CERTIFICATE

GRADE 12

INFORMATION TECHNOLOGY P1

FEBRUARY/MARCH 2014

MARKS: 120

TIME: 3 hours

This question paper consists of 15 pages and 3 annexures consisting of 4 pages in total.

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
tblRespondents.txt
tblStudents.txt

Question1_Java:
Question1.java
Question1DB.mdb
tblRespondents.txt
tblStudents.txt
TestQuestion1.java

Question2_Delphi:
DataQ2.txt
Question2P.dpr
Question2P.res
Question2U.dfm
Question2U.pas
uStudent.pas

Question2_Java:
DataQ2.txt
Student.java
TestQuestion2.java

Question3_Delphi:
DataQ3_Delphi.txt

Question3_Java:
DataQ3_Java.txt
MenuQ3_Java.txt

If you received the files above on an external medium (DVD, 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

Much research is done internationally on the use of technology. TechnoSA is a company that is busy with market research on the use of technology by South African citizens.

QUESTION 1: PROGRAMMING AND DATABASE

TechnoSA uses students as field officers who request members of the public to complete a questionnaire.

A Microsoft Office Access database named **Question1DB.mdb**, two text files named **tblRespondents.txt** and **tblStudents.txt** and an incomplete program are provided 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 for each table can be found in **ANNEXURE A**.

NOTE: Some of the data that was captured in the **tblRespondents** table is incorrect. Part of your task will be to rectify these mistakes.

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 given folder for QUESTION 1 by replacing the name of the programming language you have studied with your examination number.
- Open the given incomplete program for QUESTION 1.
- Add 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, due to 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 results for the respective queries 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 details of the respondents stored in the **tblRespondents** table, sorted by the **QuestID** field and in descending order.

Example of the output of the first six records:

NOTE: The format and order of the dates in the **DateSubmitted** field may differ from the example below.

QuestID	DateSubmitted	City	NumMobileDevices	InternetContract	ConnectionType	StudentID
30	2013/08/05	Pretoria	1	True	ADSL	8
29	2013/08/05	Port Elizabeth	2	True	ADSL	6
28	2013/08/07	Durban	1	True	ADSL	5
27	2013/08/05	Cape Town	1	False	WiMAX	8
26	2013/08/03	Johannesburg	3	True	3G	5
25	2013/08/08	East London	0	False		6

:

(3)

1.2 **Menu Option B**

Display the **QuestID**, **DateSubmitted** and **StudentID** of all the questionnaires that were submitted after 7 August 2013.

Example of output of the first six records:

NOTE: The format and order of the dates in the **DateSubmitted** field may differ from the example below.

QuestID	DateSubmitted	StudentID
1	2013/08/11	7
3	2013/08/09	4
4	2013/08/09	5
5	2013/08/10	9
6	2013/08/10	6
12	2013/08/09	5

: (3)

1.3 **Menu Option C**

Allow the user to enter the name or part of the name of a city. Display the **City**, **NumMobileDevices** and **ConnectionType** fields of respondents from the city that was entered who have an Internet contract and own two or more mobile devices.

Example of the output if the user entered the word 'town':

City	NumMobileDevices	ConnectionType
Cape Town	4	WiMAX
Cape Town	3	WiMAX

(7)

1.4 **Menu Option D**

Display a list of the cities and the average number of mobile devices owned by the respondents from each city in a calculated field named **AvgMobilePerCity**. Display the values in the calculated field with TWO decimal places.

Example of the output:

City	AvgMobilePerCity
Bloemfontein	1.00
Cape Town	2.40
Durban	1.00
East London	0.75
George	1.50
Johannesburg	1.80
Port Elizabeth	2.00
Pretoria	2.33

(5)

1.5 **Menu Option E**

Display the name, surname and current year of study of each student and the total number of questionnaires he/she submitted. Use **NumQuestionnaires** as the heading of the calculated field.

Example of the output:

Name	Surname	YearOfStudy	NumQuestionnaires
Dylan	Mullan	3	4
Gert	Smit	1	3
John	Decan	1	5
Kabelo	Mkosi	1	6
Mark	Smith	2	1
Michelle	Botha	4	2
Princess	Kweto	3	1
Sean	McCullan	2	3
Sipho	Modikaze	3	1
Thandi	Mpofu	2	4

(6)

1.6 **Menu Option F**

Increase the number of mobile devices for all the questionnaires submitted by Kabelo Mkosi by one. Once the record(s) has/have been updated successfully, a message stating that the record(s) was/were successfully processed will be displayed. The code for this message is given.

NOTE: Use Option A to verify the updating of the records for Kabelo. (5)

1.7 **Menu Option G**

Delete records from the **tblRespondents** table that do not have an Internet contract but for which a connection type is specified. Once the record(s) has/have been deleted successfully, a message stating that the record(s) was/were successfully processed will be displayed. The code for this message is given.

NOTE: Use Option A to verify the deletion of the records. (6)

- 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

TechnoSA needs to keep updated information on the students acting as field officers and monitor their productivity.

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 a text file named **DataQ2.txt** and an incomplete program that consists of:

Delphi	Java
• A class unit named **uStudent** which describes the attributes of a student and contains some methods • A main form unit named **Question2U**	• An object class named **Student** which describes the attributes of a student and contains some methods • A test class named **TestQuestion2**

The given **uStudent/Student** class contains the declaration and coding of:

- Four attributes describing a student
- Four accessor methods
- Two mutator methods

The four attributes describing a student are:

Description	Variable names used	
	Delphi	Java
Name of student	fName	name
Gender of student	fGender	gender
Number of questionnaires collected	fQuestionnaires	questionnaires
Hours spent on the survey	fHours	hours

The text file contains the data of an unknown number of students who assisted with the latest survey. The details of each student regarding the survey appears over FOUR lines of text in the file in the following format:

```
<Name of student>
<Gender of student>
<Number of completed questionnaires collected>
<Number of hours spent on distributing and collection questionnaires>
```

An example of the data for the first three students in the **DataQ2.txt** text file is given on the next page.

```
Eliana
F
30
4.25
Fairly
M
25
2.1
Neil
M
14
1.38
:
```

Do the following:

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

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

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

2.1 Do the following to complete the code in the class unit (Delphi)/object class (Java):

2.1.1 Write code for a **constructor** using parameter values to initialise the following attributes of a student:

- Name
- Gender
- Number of questionnaires collected
- Hours spent on the survey (3)

2.1.2 Write code for a method named **calcAvg** to determine the average number of questionnaires distributed and collected by a student per hour. The method returns a decimal value. (2)

2.1.3 Write code for a **toString** method that will construct and return a string which includes labels and information about a student object in the following format:

```
Student: <Name> (<Gender>)
Collected questionnaires: <Number of collected questionnaires>
Total number of hours: <Hours spent on the survey>
<Blank line>
```

Example of the output of the first two student objects using the **toString** method to display the data:

```
Student: Eliana (F)
Collected questionnaires: 30
Total number of hours: 4.25

Student: Fairly (M)
Collected questionnaires: 25
Total number of hours: 2.1

:
```

(4)

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

NOTE: An array that is capable of storing 20 student objects and a counter variable to keep track of the number of objects in the array has been declared as part of the given code.

2.2.1 Write code to read lines of text from the text file, extract the data for each student object, create the object and assign it to the array.

NOTE: The objects have to be assigned to the array before the menu options are displayed. (14)

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

Display a list of all the information of all the students using the **toString** method.

Example of the output of the data for the first two students:

```
List of students

Student: Eliana (F)
Collected questionnaires: 30
Total number of hours: 4.25

Student: Fairly (M)
Collected questionnaires: 25
Total number of hours: 2.1

:
```

(3)

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

TechnoSA wants the name of the most productive male student and the name of the most productive female student. Their productivity is measured by the average number of questionnaires they distributed and collected in an hour.

Use the **calcAvg** method to determine the two students (male and female) with the highest average number of questionnaires distributed and collected. Display the average values rounded to TWO decimal places.

Example of the output:

```
Students with the highest average values:

Male: Luca with an average of 28.46

Female: Elsa with an average of 17.86
```

(11)

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

Every week the students are required to provide the following data:

- The number of completed questionnaires they have collected
- The number of hours they spent on the survey during the week

The new data must be added to the previously recorded data for each student.

Allow the user to enter the following data from the keyboard:

- The name of the student
- The number of completed questionnaires the student collected during the week
- The number of hours the student spent on the survey during the week

NOTE: You do not have to validate the values entered by the user.

Use a conditional loop to locate the student object whose information was entered by the user in the array. If the student object is located, update the student's data in the array. If the student object is not located in the array, display a suitable message.

Test your program with the following test data sets:

Information for test data set 1:

Name: Eliana
Number of completed questionnaires collected: 17
Number of hours: 1.5

HINT: Use **Menu Option A** to display the data of all the students to see whether the student data has been updated with the relevant information.

Example of the output:

```
:

Student: Eliana (F)
Collected questionnaires: 47
Total number of hours: 5.75

:
```

Information for test data set 2:

Name: Frank
Number of completed questionnaires collected: 32
Number of hours: 3.6

Example of the output:

```
The student is not on the list.
```

(12)

- 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).

[49]

QUESTION 3: PROBLEM-SOLVING PROGRAMMING

As part of the survey, a sample of respondents was requested to provide the name of their favourite computer game and the device they use to play the game.

The information supplied by each respondent is captured as a string using the following format:

<name of game>#<device used to play the game>

The device used to play the game can be a PS3 console, an Xbox or a PC.

Examples of the strings of text that were captured:

- **Civilisation#PS3:** The game *Civilisation* is played on a PS3 console.
- **Command & Conquer#PC:** The game *Command & Conquer* is played on a PC.

The folder **Question3_XXXX** contains a text file named **DataQ3_XXXX.txt**, where XXXX refers to the programming language you have studied.

The given text file (**DataQ3_Delphi.txt** for Delphi or **DataQ3_Java.txt** for Java) contains code for the declaration and initialisation of an array named **arrData**. The array contains 35 elements of type string representing the data that was captured.

Content of the **arrData** array:

```
Civilisation#PS3; Command & Conquer#PC; Solitaire#Xbox; Chess#PC; Tetris#PC;
Chess#PC; Command & Conquer#PC; Civilisation#PC; SimCity#PC; Tetris#PC;
SimCity#PC; Civilisation#PS3; Tetris#PS3; Command & Conquer#PS3; SimCity#PC;
Solitaire#PC; Sims#Xbox; SimCity#Xbox; Command & Conquer#PC; Chess#PS3;
Tetris#Xbox; Civilisation#Xbox; SimCity#PS3; Solitaire#PC; Sims#Xbox; Command
& Conquer#PS3; Command & Conquer#PS3; Civilisation#PS3; Civilisation#PS3;
Command & Conquer#Xbox; SimCity#PS3; Solitaire#PS3; Civilisation#Xbox;
Command & Conquer#PC; SimCity#PC
```

Do the following:

- Rename the given 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 in the renamed folder for QUESTION 3.
- Add 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) by 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 two menu options: Option A and a Quit option.	• Copy the given code for displaying a menu from the **MenuQ3.txt** text file. The interface needs to display two menu options: Option A and a Quit option.
• Add a *RichEdit* component named **redQ3** on the form to display information. Change the *Align* property of the *RichEdit* component to *alClient*.	
• Copy the text for the declaration and initialisation of the **arrData** array which is supplied in the text file to your program file.	• Copy the text for the declaration and initialisation of the **arrData** array which is supplied in the text file to your program file.
• 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);`

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

3.1 Display a numbered list of the different computer games played by the respondents based on the content of the **arrData** array.

NOTE:

- The data in the given array, **arrData**, must be used to compile the list of games to display. If the content of the **arrData** array changes (for example, if another game is added), the displayed list of games must change accordingly.
- The number of different computer games in the **arrData** array is unknown.
- None of the computer games is allowed to appear more than once in the list when it is displayed. (18)

3.2 Allow the user to select a game from the numbered list (created in QUESTION 3.1) by entering the number of the game. The program must not continue unless a valid number has been entered from the list.

Use the data in the **arrData** array to calculate the following information for the selected game:

- The total number of times the game was mentioned by respondents as their favourite game
- The percentage use of each device (PS3, Xbox and PC) to play the game, rounded to ONE decimal place

Display the name of the selected game and the calculated information in the following format:

```
<name of selected game> was mentioned <number of times mentioned> times.

Percentage use of devices:
<device 1><tab><device 2><tab><device 3>
<percentage use for device 1><tab><percentage use for device 2><tab><percentage use for device 3>
```

Example 1: An example of the output if the game *Civilisation* was selected from the list:

```
Civilisation was mentioned 7 times.

Percentage use of devices:
PS3         Xbox        PC
57.1%       28.6%       14.3%
```

Example 2: An example of the output if the game *Chess* was selected from the list:

```
Chess was mentioned 3 times.

Percentage use of devices:
PS3         Xbox        PC
33.3%       0.0%        66.7%
```

(18)

- Make sure your examination number is entered as a comment in the first line of any program files containing your code.
- Save your program.
- A printout of the code will be required.

[36]

TOTAL: 120

ANNEXURE A: DATABASE STRUCTURE AND SAMPLE DATA (Page 1 of 2)

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

Table: tblRespondents

This table contains data on a survey that was done on the use of technology in South Africa.

Table structure:

Field Name	Type	Size	Description
QuestID	Number	Long Integer	A unique code assigned to each completed questionnaire
DateSubmitted	Date/Time	Date	Date on which the student submitted the questionnaire (YYYY/MM/DD)
City	Text	20	City where the respondent lives
NumMobileDevices	Number	Byte	The number of mobile devices owned by the respondent
InternetContract	Yes/No		Whether the respondent has an Internet contract, or not
ConnectionType	Text	10	The type of Internet connection the respondent uses
StudentID	Number	Long Integer	The ID code of the student who submitted the questionnaire

Sample data:

QuestID	DateSubmitted	City	NumMobileDevices	InternetContract	ConnectionType	StudentID
1	2013/08/11	Johannesburg	1	☑	3G	7
2	2013/08/01	Durban	0	☐		4
3	2013/08/09	East London	1	☐	Dial-up	4
4	2013/08/09	Durban	3	☑	3G	5
5	2013/08/10	Bloemfontein	1	☑	ADSL	9
6	2013/08/10	Johannesburg	1	☑	Dial-up	6
7	2013/08/02	East London	1	☐	3G	6
8	2013/08/07	Cape Town	4	☑	WiMAX	8
9	2013/08/04	Port Elizabeth	2	☑	ADSL	9
10	2013/08/03	Johannesburg	3	☑	ADSL	2
11	2013/08/05	Bloemfontein	1	☑	WiMAX	8
12	2013/08/09	George	2	☑	WiMAX	5
:						

NOTE: Some of the data that was captured in the **tblRespondents** table is incorrect. Part of your task will be to rectify these mistakes.

ANNEXURE A (continued) **(Page 2 of 2)**

Table: tblStudents

This table contains data on the students who conducted the survey.

Table structure:

Field Name	Type	Size	Description
StudentID	Number	Long Integer	A number to uniquely identify the student
Name	Text	20	The student's first name
Surname	Text	30	The student's surname
YearOfStudy	Number	Byte	The student's current year of study

Sample data:

StudentID	Name	Surname	YearOfStudy
1	Sipho	Modikaze	3
2	Mark	Smith	2
3	Gert	Smit	1
4	Thandi	Mpofu	2
5	John	Decan	1
6	Dylan	Mullan	3
7	Michelle	Botha	4
8	Kabelo	Mkosi	1
9	Sean	McCullan	2
10	Princess	Kweto	3

ANNEXURE B: DELPHI – TROUBLESHOOTING DATABASE PROBLEMS

B.1 If you cannot use the given database:

- Create your own database with the name **Question1DB** that includes two tables, named **tblRespondents** and **tblStudents** in the same folder as your program for QUESTION 1.
- Import the two text files (**tblRespondents.txt** and **tblStudents.txt**) to use as data for the two required tables (**tblRespondents** and **tblStudents**).
- 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 file **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 but no data is displayed:

- Click on the ADOQuery component named **qryQ1**.
- 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 **tblRespondents** and **tblStudents** in the same folder as your program for QUESTION 1.
- Import the two text files (**tblRespondents.txt** and **tblStudents.txt**) to use as data for the two required tables (**tblRespondents** and **tblStudents**).
- 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 file **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 program 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");
  }
```