



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, stiffer 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
<ul style="list-style-type: none"> • A class unit named uStudent which describes the attributes of a student and contains some methods • A main form unit named Question2U 	<ul style="list-style-type: none"> • 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
<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • 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
<ul style="list-style-type: none"> • Use a <i>MainMenu</i> component and develop an interface with two menu options: Option A and a Quit option. • Add a <i>RichEdit</i> component named redQ3 on the form to display information. Change the <i>Align</i> property of the <i>RichEdit</i> component to <i>alClient</i>. • 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; 	<ul style="list-style-type: none"> • 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. • 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: 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.

- 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.

(18)



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	<input checked="" type="checkbox"/>	3G	7
2	2013/08/01	Durban	0	<input type="checkbox"/>		4
3	2013/08/09	East London	1	<input type="checkbox"/>	Dial-up	4
4	2013/08/09	Durban	3	<input checked="" type="checkbox"/>	3G	5
5	2013/08/10	Bloemfontein	1	<input checked="" type="checkbox"/>	ADSL	9
6	2013/08/10	Johannesburg	1	<input checked="" type="checkbox"/>	Dial-up	6
7	2013/08/02	East London	1	<input type="checkbox"/>	3G	6
8	2013/08/07	Cape Town	4	<input checked="" type="checkbox"/>	WiMAX	8
9	2013/08/04	Port Elizabeth	2	<input checked="" type="checkbox"/>	ADSL	9
10	2013/08/03	Johannesburg	3	<input checked="" type="checkbox"/>	ADSL	2
11	2013/08/05	Bloemfontein	1	<input checked="" type="checkbox"/>	WiMAX	8
12	2013/08/09	George	2	<input checked="" type="checkbox"/>	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");
}
```



BYLAE C: JAVA – FOUTOPSPORING VIR DATABASISPROBLEME

C.1 Indien jy nie die gegewe databasis kan gebruik nie:

- Skip jou eie databasis met die naam **Vraag1DB** wat twee tabelle met die name **tblResponse** en **tblStudente** insluit, in dieselfde lêergids as jou program vir VRAAG 1.
- Trek ('import') die twee tekslêers **tblResponse.txt** en **tblStudente.txt** in om as data vir die twee vereiste tabelle (**tblResponse** en **tblStudente**) te gebruik.
- Die eerste reël in die tekslêers bevat die veldname wat gebruik moet word.

C.2 Indien jou program nie met die databasis kan koppel nie, maak seker dat die databasislêer **Vraag1DB** in dieselfde lêergids as jou program vir VRAAG 1 is. As dit nie die geval is nie, kopieer die databasislêer na dieselfde lêergids as jou program.

C.3 Indien jy nie met die gegewe programlêers met die databasis kan koppel nie, gebruik die volgende bronkode om databasiskonnektiwiteit te verseker:

```

try
{
    Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
    String filename = "Vraag1DB.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 ("Nie in staat om met die databasis te koppel nie");
}
    
```



BYLAE B: DELPHI – FOUTOPSPORING VIR DATABASISPROBLEME

- B.1 Indien jy nie die gegewe databasis kan gebruik nie:
- Skip jou eie databasis met die naam **Vraag1DB** wat twee tabelle met die name **tblResponse** en **tblStudente** insluit, in dieselfde lêergids as jou program vir VRAAG 1.
 - Trek ('Import') die twee tekslêers (**tblResponse.txt** en **tblStudente.txt**) in om as data vir die twee vereiste tabelle (**tblResponse** en **tblStudente**) te gebruik.
 - Die eerste reël in die tekslêers bevat die veldname wat gebruik moet word.
- B.2 Indien jou program nie met die databasis kan koppel nie, maak seker dat die databasislêer **Vraag1DB** in dieselfde lêergids as jou program vir VRAAG 1 is. As dit nie die geval is nie, kopieer die databasislêer na dieselfde lêergids as jou program.
- B.3 Indien jou program met die databasis kan koppel, maar geen data word vertoon nie:
- Klik op die 'ADOQuery'-komponent met die naam **qryVR1**.
 - Klik op die ellipsknoppie (drie kolleëies) regs van die 'ConnectionString'-eienskap in die 'Object Inspector'.
 - Klik op die 'Build'-knoppie, wat jou na die 'Data Link Properties'-dialoogvenster lei.
 - Klik op die 'Provider tab' om die 'Provider' te kies en kies 'Microsoft Jet 4.0 OLE DB Provider'. Klik op die 'Next'-knoppie.
 - Die 'Connection tab' sal vertoon word. Die eerste opsie op die 'Connection tab' bevat 'n ellipsknoppie (drie kolleëies) wat jou toelaat om deur te blaai en die **Vraag1DB**-lêer te soek. Jy sal hierdie lêer in die **Vraag1**-lêergids ('folder') kry. Sodra jy dit gekry het, kies die **Vraag1DB**-lêer en klik dan op die 'Open'-knoppie.
 - Verwyder die gebruikersnaam ('username'), 'Admin'.
 - Klik op die 'Test Connection'-knoppie.
 - Klik 'OK' op elk van die oop dialoogvensters.



BYLAE A (vervolg)

(Bladsy 2 van 2)

Tabel: tblStudente

Hierdie tabel bevat data oor die studente wat die opname gedoen het.

Tabelstruktuur:

Field Name	Type	Size	Description
StudentID	Number	Long Integer	'n Nommer om elke student uniek te identifiseer
Naam	Text	20	Die student se noernaam
Van	Text	30	Die student se van
JaarVanStudie	Number	Byte	Die student se huidige studiejaar

Voorbeeldata:

StudentID	Naam	Van	JaarVanStudie
1	Sipho	Modikaze	3
2	Mark	Smith	2
3	Gert	Smit	1
4	Thandi	Mpotu	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



BYLAE A: DATABASISSTUKTUR EN VOORBEELDDATA (Bladsy 1 van 2)

Hierdie bylae vertoon die datastruktuur en voorbeelddata vir die tabelle wat in die **Vraag1DB**-databasis in **VRAAG 1** gebruik word.

Tabel: tblRespondente

Hierdie tabel bevat data oor 'n opname wat oor die gebruik van tegnologie in Suid-Afrika gedoen is.

Tabelstruktuur:

Field Name	Type	Size	Description
VraeysID	Number	Long Integer	'n Unieke kode wat aan elke voltooiide vraeys toegeken word
DatumIngegee	Date/Time	Date	Datum waarop die student die vraeys ingelewer het
Stad	Text	20	Stad waar die respondent woon
GetalMobileToestelle	Number	Byte	Die getal mobile toestelle wat die respondent besit
InternetKontrak	Yes/No	10	Of die respondent 'n Internetkontrak het, of nie
KonneksieTipe	Text		Die tipe Internetkonneksie wat die respondent gebruik
StudentID	Number	Long Integer	Die ID-kode van die student wat die vraeys ingelewer het

Voorbeelddata:

VraeysID	DatumIngegee	Stad	GetalMobileToestelle	InternetKontrak	KonneksieTipe	StudentID
1	2013/08/11	Johannesburg	1	<input checked="" type="checkbox"/>	3G	7
2	2013/08/01	Durban	0	<input type="checkbox"/>		4
3	2013/08/09	Oos-Londen	1	<input type="checkbox"/>	Skakellyn	4
4	2013/08/09	Durban	3	<input checked="" type="checkbox"/>	3G	5
5	2013/08/10	Bloemfontein	1	<input checked="" type="checkbox"/>	ADSL	9
6	2013/08/10	Johannesburg	1	<input checked="" type="checkbox"/>	Skakellyn	6
7	2013/08/02	Oos-Londen	1	<input type="checkbox"/>	3G	6
8	2013/08/07	Kaapstad	4	<input checked="" type="checkbox"/>	WIMAX	8
9	2013/08/04	Port Elizabeth	2	<input checked="" type="checkbox"/>	ADSL	9
10	2013/08/03	Johannesburg	3	<input checked="" type="checkbox"/>	ADSL	2
11	2013/08/05	Bloemfontein	1	<input checked="" type="checkbox"/>	WIMAX	8
12	2013/08/09	George	2	<input checked="" type="checkbox"/>	WIMAX	5

LET WEL: Sommige van die data wat in die **tblRespondente**-tabel ingesleutel is, is foutief. Deel van jou opdrag sal wees om hierdie foute te korrigeer.



Gebruik die data in die **arrData**-skikking om die volgende inligting vir die spelletjie wat gekies is, te bereken:

- Die totale getal kere wat die spelletjie deur respondente as hulle gunstelingspelletjie genoem is
- Die persentasie gebruik van elke toestel (PS3, Xbox en PR ('PC')) om die spelletjie te speel, afgerond tot EEN desimale plek

Vertoon die naam van die spelletjie wat gekies is en die inligting wat bereken is in die volgende formaat:

```
<naam van spelletjie wat gekies is> is <getal keer genoem> keer genoem.
Persentasie gebruik van toestelle:
<toestel 1><<toestel 2><<toestel 3>
<persentasie gebruik van toestel 1><<persentasie gebruik van toestel 2>
<tab><persentasie gebruik van toestel 3>
```

Voorbeeld 1: 'n Voorbeeld van die afvoer indien die spelletjie *Civilisation* uit die lys gekies is:

```
Civilisation is 7 keer genoem.
Persentasie gebruik van toestelle:
PS3 Xbox PR
57.1% 28.6% 14.3%
```

Voorbeeld 2: 'n Voorbeeld van die afvoer indien die spelletjie *Chess* uit die lys gekies is:

```
Chess is 3 keer genoem.
Persentasie gebruik van toestelle:
PS3 Xbox PR
33.3% 0.0% 66.7%
```

(18)

- Maak seker dat jou eksamennummer as 'n kommentaar in die eerste reël van enige program lêer wat jou kode bevat, ingesluit is.
- Stoor jou program.
- 'n Drukstuk van die kode sal vereis word.

[36]

TOTAAL:

120



- Ontwikkel 'n koppelvlak soos volg:

Delphi	Java
<ul style="list-style-type: none"> • Gebruik 'n <i>MainMenu</i>-komponent en ontwikkel 'n koppelvlak met twee keuselypsies: Opsie A en 'n Verlaat-opsie. • Voeg 'n <i>RichEdit</i>-komponent met die naam redV3 op die vorm om inligting te vertoon. Verander die <i>Align</i>-eienskap van die <i>RichEdit</i>-komponent na <i>align</i>. • Kopieer die teks vir die verklaring en inisialisering van die <i>arrData</i>-skikking wat in die teksleër verskat is, na jou program. • Gebruik die volgende kode om die Verlaat-keuselypsie te voltooi: <code>Application.Terminate;</code> 	<ul style="list-style-type: none"> • Kopieer die gegewe kode om 'n keuselys uit die MenuV3.txt-teksleër te vertoon. Die koppelvlak moet twee keuselypsies vertoon: Opsie A en 'n Verlaat-opsie. • Kopieer die teks vir die verklaring en inisialisering van die <i>arrData</i>-skikking wat in die teksleër verskat is, na jou program. • Gebruik die volgende kode om die Verlaat-keuselypsie te voltooi: <code>System.exit(0);</code>

Skryf kode om **Keuselypsie A** te voltooi om die volgende te doen:

- 3.1 Vertoon 'n genommerde lys van die verskillende rekenaarspeletjies wat deur die respondente gespeel word, gebaseer op die inhoud van die *arrData*-skikking.

LET WEL:

- Die data in die gegewe skikking, *arrData*, moet gebruik word om die lys van speletjies wat vertoon moet word, saam te stel. As die inhoud van die *arrData*-skikking verander (byvoorbeeld, as 'n ander speletjie bygevoeg word), moet die lys van speletjies wat vertoon word, dienoreenkomstig verander.
- Die aantal verskillende rekenaarspeletjies in die *arrData*-skikking is onbekend.
- Geen rekenaarspeletjie mag meer as een keer op die lys verskyn wanneer dit vertoon word nie.

(18)

- 3.2 Laat die gebruiker toe om 'n speletjie uit die genommerde lys (wat in VRAAG 3.1 geskep is) te kies deur die nommer van die speletjie in te sleutel. Die program moet nie voortgaan voordat 'n geldige nommer uit die lys ingesleutel is nie.



VRAAG 3: PROBLEEMOPLOSSINGSPROGRAMMERING

As deel van die opname is 'n groep respondente versoek om die naam van hulle gunstelingrekenaarspeletjie en die toestel wat hulle gebruik om die speletjie mee te speel, te verskat.

Die inligting wat deur elke respondent verskat is, is as 'n string in die volgende formaat vasgelê:

```
<naam van speletjie>#<toestel wat gebruik word om speletjie mee te speel>
```

Die toestel wat gebruik word om die speletjie mee te speel, kan 'n PS3-konsole, 'n Xbox of 'n PR ('PC') wees.

Voorbeelde van die stringe teks wat vasgelê is:

- **Civilisation#PS3:** Die *Civilisation*-speletjie word op 'n PS3-konsole gespeel.
- **Command & Conquer#PR:** Die *Command & Conquer*-speletjie word op 'n PR ('PC') gespeel.

Die lêergids **Vraag3_XXXX** bevat 'n teks lêer met die naam **DataV3_XXXX.txt**, waar XXXX verwys na die programmeringstaal wat jy bestudeer het.

Die gegewe teks lêer (**DataV3_Delphi.txt** vir Delphi of **DataV3_Java.txt** vir Java) bevat kode vir die verklaaring en inisialisering van 'n skikking met die naam **arrData**. Die skikking bevat 35 elemente van tipe string wat die data wat ingesleutel is, voorstel.

Inhoud van die **arrData**-skikking:

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

Doen die volgende:

- Herneem die gegewe lêergids vir VRAAG 3 deur die naam van die programmeringstaal wat jy bestudeer het, met jou eksamennummer te vervang.
- Skep 'n nuwe program/projek/toepassing in die herneemde lêergids vir VRAAG 3.
- Sleutel jou eksamennummer in as 'n kommentaarstelling in die eerste reël van die program lêer(s) wat jy geskep het en wat jou kode sal bevat.
- Stoor die program lêer(s) deur die **vraagnummer** as deel van die lêernaam in die herneemde lêergids vir VRAAG 3 te gebruik.



Toets jou program met die volgende stelle toetsdata:

Inligting vir toetsdatastel 1:

Naam: Eliana
 Getal voltooidde vrae/lyste wat ingesamel is: 17
 Getal ure: 1.5

WENK: Gebruik **Keuselysopsie A** om die data van al die studente te vertoon om te sien of die student se data met die relevante inligting opgedateer is.

Voorbeeld van die afvoer:

```

:
Student: Eliana (V)
Ingesamelde vrae/lyste: 47
Totale getal ure: 5.75
:
    
```

Inligting vir toetsdatastel 2:

Naam: Frank
 Getal voltooidde vrae/lyste wat ingesamel is: 32
 Getal ure: 3.6

Voorbeeld van die afvoer:

```

Die student is nie op die lys nie.
    
```

(12)

Maak seker dat jou eksamennummer as 'n kommentaar in die eerste reël van die klaseenheid ('class unit') (Delphi)/objekklas ('object class') (Java) asook die hoofvormeenheid ('main form unit') (Delphi)/toetsklas ('test class') (Java) ingesleutel is.

- Stoor al die lêers.
- Maak 'n drukstuk van die kode indien dit vereis word. Druk die klaseenheid ('class unit') (Delphi)/objekklas ('object class') (Java) en die hoofvormeenheid ('main form unit') (Delphi)/toetsklas ('test class') (Java).

[49]



2.2.3

Skryf kode om **Kenselysopsie B** te voltooi om die volgende te doen:

TechnoS A wil die naam van die mees produktiewe manlike student en die naam van die mees produktiewe vroulike student hê. Hulle produktiwiteit word gemeet deur die gemiddelde getal vraelyste wat hulle in 'n uur versprei en ingesamel het.

Gebruik die **berekenGem**-metode om die twee studente (manlik en vroulik) te bepaal wat die hoogste gemiddelde getal vraelyste versprei en ingesamel het. Vertoon die gemiddelde waardes, afgerond tot TWEE desimale plekke.

Voorbeeld van die afvoer:

Studente met die hoogste gemiddelde waardes: Manlik: Luca met 'n gemiddeld van 28.46 Vroulik: Elsa met 'n gemiddeld van 17.86

(11)

2.2.4

Skryf kode om **Kenselysopsie C** te voltooi om die volgende te doen:

Elke week word daar van die studente vereis om die volgende data te verskat:

- Die getal voltooië vraelyste wat hulle ingesamel het
- Die getal ure wat hulle gedurende die week aan die opname spandeer het

Die nuwe data moet bygevoeg word by die data wat voorheen vir elke student aangeteke is.

Laat die gebruiker toe om die volgende data vanaf die sleutelbord in te sleutel:

- Die naam van die student
- Die getal voltooië vraelyste wat die student gedurende die week ingesamel het
- Die getal ure wat die student gedurende die week aan die opname spandeer het

LET WEL: Jy hoef nie die data te valideer wat deur die gebruiker ingesleutel is nie.

Gebruik 'n voorwaardelike lus om die student-objek met inligting wat deur die gebruiker ingesleutel is, in die skikking op te spoor. Indien die student-objek opgespoor word, moet jy die student se data in die skikking opdateer. Indien die student se data nie in die skikking opgespoor word nie, vertoon 'n gepaste boodskap.



2.1.3

Skryf kode vir 'n **toString**-metode wat 'n string wat byskrifte en inligting oor 'n student-objek insluit, in die volgende formaat sal saamstel en terugstuur:

```
Student: <Naam> (<Geslag>)
Ingesamelde vraelyste: <Getal vraelyste ingesamel>
Totaal getal ure: <Ure wat aan die opname spandeer is>
<Opp reël>
```

Voorbeeld van die afvoer van die eerste twee student-objekte deur die **toString**-metode te gebruik om die data te vertoon:

```
Student: Eliana (V)
Ingesamelde vraelyste: 30
Totaal getal ure: 4.25

Student: Fairly (M)
Ingesamelde vraelyste: 25
Totaal getal ure: 2.1
```

(4)

2.2

Doen die volgende om die kode in die hoofvormeenheid ('main form unit') (Delphi)/toetsklas ('test class') (Java) te voltooi:

LET WEL: 'n Skikking wat in staat is om 20 student-objekte te stoor en 'n tellerveranderlike om tred te hou met die getal objekte in die skikking is as deel van die gegewe kode verklaar.

2.2.1

Skryf kode om reëls teks uit die teksleër te lees, die data vir elke student-objek te onttrek, die objek te skep en dit aan die skikking toe te ken.

LET WEL: Die objekte moet aan die skikking toegeken word voordat die keuselypsies vertoon word.

(14)

2.2.2

Skryf kode om **Keuselypsie A** te voltooi om die volgende te doen:

Vertoon 'n lys van al die inligting van al die studente deur die **toString**-metode te gebruik.

Voorbeeld van die afvoer van die data vir die eerste twee studente:

```
Lys van studente

Student: Eliana (V)
Ingesamelde vraelyste: 30
Totaal getal ure: 4.25

Student: Fairly (M)
Ingesamelde vraelyste: 25
Totaal getal ure: 2.1
```

(3)



Elizama	V	30	4.25	Fairly	M	25	2.1	Netl	M	14	1.38	:
---------	---	----	------	--------	---	----	-----	------	---	----	------	---

Doen die volgende:

- Herneem die gegewe lêrigids vir VRAAG 2 deur die naam van die programmeringstaal wat jy bestudeer het, met jou eksamennummer te vervang.

Delphi	Java
<ul style="list-style-type: none"> • Maak die gegewe onvtooid programmeer Vraag2P.dpr oop. • Maak die onvtooid Student.pas-klaseenheid oop. • Voeg jou eksamennummer as 'n kommentaar in die eerste reël van beide die Student.pas-klaseenheid en die Vraag2U.pas-hoofvormeenheid in. 	<ul style="list-style-type: none"> • Maak die gegewe onvtooid toetsklas ToetsVraag2.java oop. • Maak die onvtooid Student.java oop. • Voeg jou eksamennummer as 'n kommentaar in die eerste reël van beide die Student.java-objekklas en die ToetsVraag2.java-toetsklas in.

- Kompileer en voer die program uit. Die koppelvlak vertoon vier keuselysopsies: Opsie A tot Opsie C en 'n Verlaat-opsie.

2.1 Doen die volgende om die kode in die klaseenheid ('class unit') (Delphi)/objekklas ('object class') (Java) te voltooi:

2.1.1 Skryf kode vir 'n **konstruktor** ('constructor') wat parameterwaardes gebruik om die volgende attribute van 'n student te inisialiseer:

- Naam
- Geslag
- Getal vraelyste ingesamel
- Ure wat aan die opname spandeer is

2.1.2 Skryf kode vir 'n metode met die naam **berekenGem** om die gemiddelde getal vraelyste wat 'n student per uur versprei en ingesamel het, te bereken. Die metode stuur 'n desimale waarde terug.

(2)

(3)



VRAAG 2: OBJEK-GEORIENTEERDE PROGRAMMERING

TechnoSA moet opgedateerde inligting hou van die studente wat as veldwerkers optree en hul produktiwiteit monitor.

Die lêers wat vir hierdie vraag benodig word, kan in die lêergids met die naam **Vraag2_XXXX** gevind word, waar XXXX verwys na die programmeringstaal wat jy bestudeer het.

Jy is voorsien van 'n tekslêer met die naam **DataV2.txt** en 'n onvolledige program wat bestaan uit:

Delphi	<ul style="list-style-type: none"> • 'n Klaseenheid ('class unit') met die naam Student wat die attribute van 'n student beskryf en 'n paar metodes bevat • 'n Hoofvormeenheid ('main form unit') met die naam Vraag2
Java	<ul style="list-style-type: none"> • 'n Objekklas ('object class') met die naam Student wat die attribute van 'n student beskryf en 'n paar metodes bevat • 'n Toetsklas ('test class') met die naam ToetsVraag2

Die gegee **uStudent/Student**-klas bevat die verklaring en kodering van:

- Vier attribute wat 'n student beskryf
- Vier haal ('accessor')-metodes
- Twee wysigings ('mutator')-metodes

Die vier attribute wat 'n student beskryf, is:

Beskrywing	
Veranderlike name wat gebruik is	Delphi
Naam van student	fNaam
Geslag van student	fGeslag
Getal vraelyste ingesamel	fVraelyste
Ure wat aan die opname spandeer is	fUre

Die tekslêer bevat die data van 'n onbekende getal studente wat met die jongste opname gehelp het. Die besonderhede van elke student met betrekking tot die opname strek oor VIER teksreëls in die lêer in die volgende formaat:

<Naam van student>
<Geslag van student>
<Getal voltooiëde vraelyste ingesamel>
<Getal ure wat aan die verspreiding en insameling van vraelyste spandeer is>

'n Voorbeeld van die data vir die eerste drie studente in die **DataV2.txt**-tekslêer word op die volgende bladsy gegee.



1.6

Kenselopsie F

Vermeerder die getal mobiele toestelle vir al die vraelyste wat deur Kabelo Mkosi ingelewer is met een. Sodra die rekord(s) suksesvol opgedateer is, sal 'n boodskap wat sê dat die rekord(s) suksesvol verwerk is, vertoon word. Die kode vir hierdie boodskap is gegee.

LET WEL: Gebruik Opsie A om die opdatering van die rekords vir Kabelo te verifieer.

(5)

1.7

Kenselopsie G

Verwyder rekords uit die **tblRespondente**-tabel wat nie 'n Internetkontrak het nie, maar waarvoor daar wel 'n tipe konneksie gespesifiseer is. Sodra die rekord(s) suksesvol verwyder is, sal 'n boodskap wat sê dat die rekord(s) suksesvol verwerk is, vertoon word. Die kode vir hierdie boodskap word gegee.

LET WEL: Gebruik Opsie A om die verwydering van die rekords te verifieer.

(6)

- Sleutel jou eksamennummer in as 'n kommentaar in die eerste reël van die lêer wat die SQL-stellings bevat.
- Stoor jou program.
- Maak 'n drukstuk van die kode indien dit vereis word.

[35]



1.3

Kueslipsis C

Laat die gebruiker toe om 'n stad se naam of 'n gedeelte van die stad se naam in te sleutel. Vertoon die **Stad**, **GetalMobileToestelle**- en **KonneksieTipe**-velde van respondente uit die stad wat ingesleutel is wat 'n Internetkontrak het en twee of meer mobile toestelle besit.

Voorbeeld van die afvoer indien die gebruiker die woord 'stad' insleutel:

Stad	GetalMobileToestelle	KonneksieTipe
Kaapstad	4	WIMAX
Kaapstad	3	WIMAX

(7)

1.4

Kueslipsis D

Vertoon 'n lys van die stede en die gemiddelde getal mobile toestelle wat die respondente van elke stad besit in 'n berekende veld met die naam **GemobilePerStad**. Vertoon die waardes in die berekende veld met TWEE desimale plekke.

Voorbeeld van die afvoer:

Stad	GemobilePerStad
Bloemfontein	1.00
Durban	1.00
George	1.50
Johannesburg	1.80
Kaapstad	2.40
Oos-Londen	0.75
Port Elizabeth	2.00
Pretoria	2.33

(5)

1.5

Kueslipsis E

Vertoon die naam, van en die huidige studiejaar van elke student en die totale getal vrae wat hy/sy ingelewer het. Gebruik **GetalVrae** as die opschrift van die berekende veld.

Voorbeeld van die afvoer:

Naam	Van	JaarVanStudie	GetalVrae
Dylan	Mulan	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	Mpotu	2	4

(6)



- Voortoi die kode vir elke keuselysopsie deur 'n toepaslike SQL-stelling te formuleer om die resultaat van die onderskeie navrae ('queries') te vertoon, soos in VRAAG 1.1 tot 1.7 hieronder beskryf word.

LET WEL: Die kode van sommige toevoerstellings en die kode om die SQL-stellings uit te voer en die resultate van die navrae te vertoon, is reeds as deel van die gegewe kode geskryf.

1.1 Keuselysopsie A

Vertoon al die besonderhede van die respondente wat in die **tblRespondente**-tabel gestoor is, in dalende volgorde en volgens die **VraeysID**-veld gesorteer.

Voorbeeld van die afvoer van die eerste ses rekords:

LET WEL: Die formaat en volgorde van die datums in die **Datumingegee**-veld mag van die voorbeeld hieronder verskil.

VraeysID	Datumingegee	Stad	GetalMobileToestelle	InternetKontrak	KonneksieTipe	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	Kapstad	1	False	WiMAX	8
26	2013/08/03	Johannesburg	3	True	3G	5
25	2013/08/08	Oos-Londen	0	False		6

(3)

1.2 Keuselysopsie B

Vertoon die **VraeysID**, **Datumingegee** en **StudentID** van al die vraeyste wat na 7 Augustus 2013 ingelewer is.

Voorbeeld van afvoer van die eerste ses rekords:

LET WEL: Die formaat en volgorde van die datums in die **Datumingegee**-veld mag van die voorbeeld hieronder verskil.

VraeysID	Datumingegee	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)



SCENARIO

Baie navorsing word internasionaal oor die gebruik van tegnologie gedoen. TechnoSA is 'n maatskappy wat besig is met marknavorsing oor Suid-Afrikaanse burgers se gebruik van tegnologie.

VRAAG 1: PROGRAMMERING EN DATABASIS

TechnoSA gebruik studente as veldwerkers wat lede van die publiek versoek om 'n vraelys te voltooi.

'n Microsoft Office Access-databasis met die naam **Vraag1DB.mdb**, twee tekslêers genaamd **tblRespondente.txt** en **tblStudente.txt** en 'n onvoltooid program word verskat in die lêrgids met die naam **Vraag1_XXXX**, waar **XXXX** verwys na die programmeringstaal wat jy bestudeer het.

Die ontwerp van die tabelle in die **Vraag1DB**-databasis en voorbeelddata vir elke tabel kan in **BYLAAG A** gevind word.

LET WEL: Sommige van die data wat in die **tblRespondente**-tabel ingesleutel is, is foutief. Deel van jou opdrag sal wees om hierdie foute te korrigeer.

Doen die volgende:

- Maak 'n rugsteunkopie van die **Vraag1DB**-databasis VOORDAT jy begin om die vrae te beantwoord. Jy sal 'n kopie van die oorspronklike databasis nodig hê om jou program deeglik te kan toets.

- Herneem die gegewe lêrgids vir **VRAAG 1** deur die naam van die programmeringstaal wat jy bestudeer het, met jou eksamennummer te vervang.

- Maak die gegewe onvoltooid program vir **VRAAG 1** oop.

- Voeg jou eksamennummer as 'n kommentaar in die eerste reël van die programlêer by.

- Kompilleer en voer die program uit. Die koppelvlak vertoon agt keuselysopsies: Opsie A tot Opsie G, en 'n Verlaat-opsie.

LET WEL:

- As gevolg van die onvoltooid SQL-stellings, sal 'n foutboodskap vertoon word indien enige van Opsie A tot Opsie G gekies word.

- Indien jy enige probleme ondervind om die databasis te gebruik of om met die databasis te koppel, verwys na **BYLAAG B (Delphi)/BYLAAG C (Java)** vir foutopsortingswenke.

- Indien jy steeds databasisprobleme ondervind, moet jy nogtans die SQL-kode doen en dit inlewer om nagesien te word. **Punte sal siegs toegeken word vir die programmeringskode wat die SQL-stellings bevat.**



8. Stoor jou werk met gereelde tussenposes as 'n voorsorgmaatreeël teen kragonderbrekings.
9. Gedurende die eksamen mag jy die handleidings gebruik wat oorspronklik saam met die apparaat en programmatuur verskat is. Jy mag ook die HELP-funksies van die programmatuur gebruik. Java-kandidate mag die Java API-lêers gebruik. Jy mag NIE enige ander hulpbronnemateriaal gebruik NIE.
10. Aan die einde van hierdie eksamensessie moet jy die ekssterne medium met al jou werk daarop gestoor, inlewer OF jy moet seker maak dat al jou werk op die netwerk gestoor is, soos deur die toetsinghouer/onderwyser aan jou verduidelik is.
11. Maak seker dat alle lêers wat op die ekssterne medium of netwerk gestoor is, gelees kan word.
12. Indien dit vereis word, maak drukstukke van die programmeringskode van al die vrae wat jy gedoen het.
13. Alle drukwerk van die vrae wat jy gedoen het, sal binne een uur nadat hierdie eksamen afgehandel is, plaasvind.



INSTRUKSIES EN INLIGTING

1. Die duur van hierdie eksamen is drie uur. As gevolg van die aard van hierdie eksamen is dit belangrik om kennis te neem dat jy nie toegelaat sal word om die eksamenlokaal voor die einde van die eksamensessie te verlaat nie.
2. Geen onderskeid is in hierdie vraestel getref tussen die twee programmeringstale met betrekking tot die formulering van die vrae oor programmering nie. Waar nodig, word spesifieke instruksies vir onderskeidelik Delphi- en Java-kandidate gegee.
3. Jy het die lys lêers hieronder nodig om die vrae te beantwoord. Die toetsighouer/onderwyser sal vir jou sê waar om dit te vind.

Vraag1_Delphi:

tblResponse.txt
tblStudente.txt
Vraag1_DB.mdb
Vraag1_P.dpr
Vraag1_P.res
Vraag1_U.dfm
Vraag1_U.pas

Vraag1_Java:

tblResponse.txt
tblStudente.txt
ToetsVraag1.java
Vraag1.java
Vraag1_DB.mdb

Vraag2_Delphi:

DataV2.txt
uStudent.pas
Vraag2_P.dpr
Vraag2_P.res
Vraag2_U.dfm
Vraag2_U.pas

Vraag2_Java:

DataV2.txt
Student.java
ToetsVraag2.java

Vraag3_Delphi:

DataV3_Delphi.txt

Vraag3_Java:

DataV3_Java.txt
KeuseV3_Java.txt

Indien jy die lêers hierbo op 'n eksterne medium (DVD, CD, stiftie of fliksaandrywer ('flash drive')) ontvang het, skryf jou eksamennummer op die etiket.

4. Tik jou eksamennummer as 'n kommentaar in die eerste reël van elke programlêer wat jou programmeringskode bevat.

5. Jou program moet altyd gekodeer word om die vraag op so 'n manier te beantwoord dat dit met verskillende stalle toeverdata uitgevoer kan word.

6. Lees AL die vrae aandagtig deur. Moenie meer doen as wat die vrae vereis nie.

7. Lees die hele vraag voordat jy enige subvrae beantwoord.





Hierdie vraestel bestaan uit 15 bladsye en 3 bylaes wat
altesaam uit 4 bladsye bestaan.

TYD: 3 uur

PUNTE: 120

INLIGTINGSTEGNOLOGIE V1
FEBRUARIE/MART 2014

GRAAD 12

NASIONALE
SENIOR SERTIFIKAT



Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

basic education