



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NATIONAL SENIOR CERTIFICATE

GRADE 12

INFORMATION TECHNOLOGY P1

FEBRUARY/MARCH 2012

MARKS: 120

TIME: 3 hours



This question paper consists of 31 pages, 3 addenda and an information sheet.



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. Answer SECTION A (for Delphi programmers) OR SECTION B (for Java programmers).
3. You require the files listed below in order to answer the questions. They are EITHER on a stiffy disk OR CD issued to you OR the invigilator/teacher will tell you where to find them on the hard drive of the workstation you are using OR in which network folder.

QUESTION 1**Delphi:**

FundsDB.mdb
Question1_P.dpr
Question1_P.res
Question1_U.dfm
Question1_U.pas
tblDonations.txt
tblStalls.txt

Java:

Funds.java
FundsDB.mdb
tblDonations.txt
tblStalls.txt
TestQuestion1.java

QUESTION 2**Delphi:**

Data2011.txt
Question2_P.dpr
Question2_P.res
Question2_U.dfm
Question2_U.pas
uEvent.pas

Java:

Data2011.txt
Event.java
TestQuestion2.java

QUESTION 3**Delphi:**

Question3_P.dpr
Question3_P.res
Question3_U.dfm
Question3_U.pas

Java:

TestQuestion3.java

If you received a disk (CD or stiffy) containing the files above, write your examination number on the label.

4. Save your work at regular intervals as a precaution against power failures.
5. Save ALL your solutions in folders with the question number and your examination number as the name of the folder, for example Quest2_3020160012.



6. Type in your examination number as a comment in the first line of each program.
7. Read ALL the questions carefully. Do not do more than the questions require.
8. 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.**
9. At the end of this examination session you must hand in the disk or CD with all your work saved on it OR you must make sure that all your work has been saved on the hard drive/network as explained to you by the invigilator/teacher. Ensure that all files can be read.
10. The invigilator will inform you whether you should hand in printouts of the programming code of all the questions you have done.
11. If printing is required, all printing of programming questions that you have done will take place within an hour of the completion of this question paper.
12. Complete the information sheet attached to this question paper and hand it in at the end of this examination session.



SECTION A

Answer ALL the questions in this section only if you studied **Delphi**.

SCENARIO

Fund-raising has become an integral part of an annual school programme. The community, together with the teachers and learners at Sundowns High School, put their efforts together in organising several events throughout the year to ensure that the school has sufficient funds to support the school's academic, sporting and cultural activities.

QUESTION 1: DELPHI PROGRAMMING AND DATABASE

One fundraising event the school decided to have is a curry and rice day. Mr Sweetie, the fundraising coordinator at the school, has created a database named **FundsDB** to store the data for the curry and rice day.

Each class is responsible for a stall that will be set up, either on rugby fields A, B or C or on hockey fields A, B or C. Space is allocated to each stall according to the estimated number of guests that will visit the stall. Donations are collected towards the purchasing of some of the requirements (meat, rice, vegetables or other items) that are needed to prepare and serve the curry and rice.

An incomplete program has been developed to process queries on the data in the given database. Your task will be to complete this program.

The database named **FundsDB**, as well as an incomplete Delphi project named **Question1_P.dpr**, has been saved in the folder named **Question1_Delphi**.

NOTE: The design of the tables in the **FundsDB** database and the sample data for this question can be found in **ADDENDUM A: Table Description Sheet**.

NOTE: If you cannot use the database provided, follow the instructions in **ADDENDUM B** to create the database before you answer any of QUESTIONS 1.1 to 1.7.

NOTE: Make a copy of the **FundsDB** database BEFORE you start with the solution. You will need a copy of the original database to be able to test your program thoroughly.

Do the following:

- Rename the folder **Question1_Delphi** as **Question1_X**, where X should be replaced with your examination number.
- Open Delphi and then open the file **Question1_P.dpr** in the folder **Question1_X**. The program displays eight buttons as well as a DBGrid that will be used as an output component (see example on the next page).
- Add your examination number to the right of 'Question 1 –' in the caption of the form.



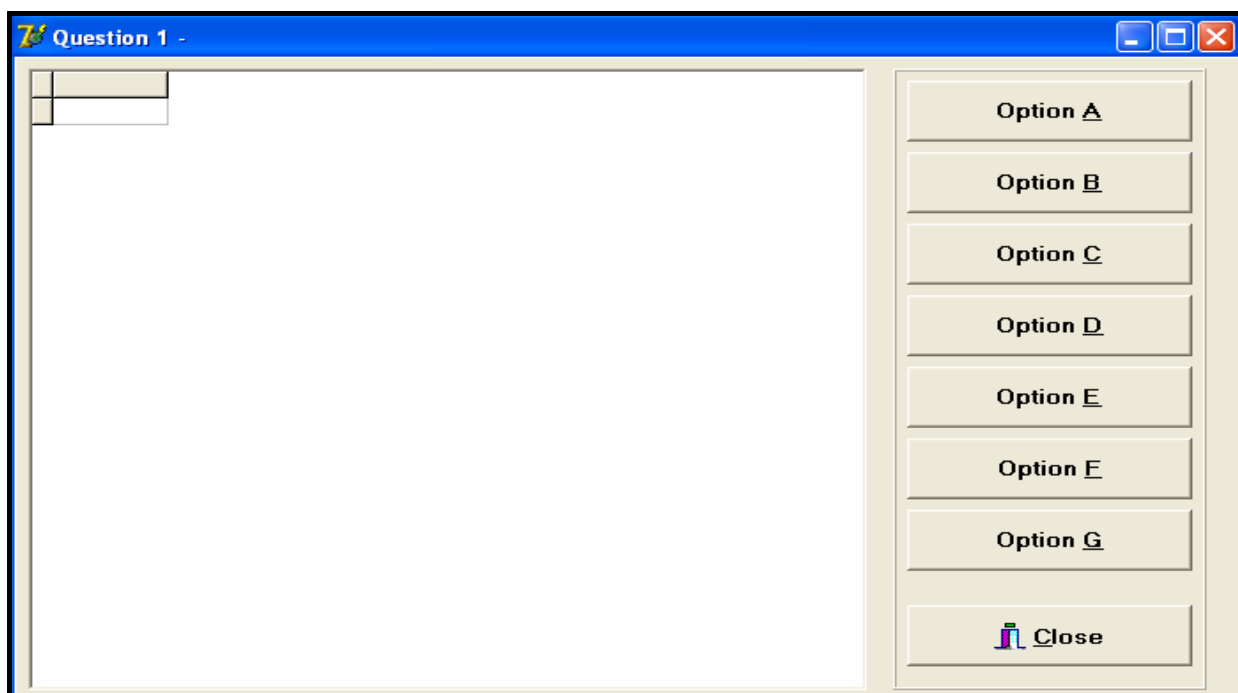
- Go to 'File/Save As ...' and save the unit as **Question1_UXXXX** (where XXXX must be replaced with the last FOUR digits of your examination number).
- Go to 'File/Save Project As ...' and save the project as **Question1_PXXXX** (where XXXX must be replaced with the last FOUR digits of your examination number).
- The program should be able to connect to the database named **FundsDB**. When you do QUESTION 1.1 (on the next page) and you find that the connectivity is not in place, use the steps supplied in **ADDENDUM C** to establish connection with the database.

NOTE: If your program cannot connect to the database, make sure that the database file **FundsDB** is in the same folder as your program. Your program will not work if the database file is in a folder other than the folder containing the program. If this is the case, copy the database file **FundsDB** into the same folder as your program.

NOTE: If you still cannot establish connectivity with the database when you execute the program, you must still do the SQL code and submit it for marking.

Marks will only be awarded for the programming code that contains the SQL statements in the unit named Question1_UXXXX.

When you execute the program, the interface below will be displayed. An error will be displayed when you click the buttons, due to the incomplete SQL statements.



Do the following:

Complete the SQL statements in **Question1_UXXXX.pas** for each button, as indicated in QUESTIONS 1.1 to 1.7 below. The code to execute the SQL statements and to display the results in the DBGrid has been given to you. You only need to complete the SQL statements and some input statements, as required in the **Question1_UXXXX** unit.

- 1.1 Mr Sweety wants a list of the estimated number of guests that will visit the stalls. Complete the code for the **Option A** button by formulating an SQL statement to display the **stall name**, **class** and **number of guests** for all the stalls in the **tblStalls** table. Display the results according to the number of guests in descending order.

Example of the output for the first five records:

StallName	Class	NumOfGuests
Crazy Dolls	9B	144
Gaming Palace	8A	142
The Rock	11A	136
The 'WW'	9C	126
Double Espresso	9E	122

:

(3)

- 1.2 Miss Shelby, the deputy principal, needs to contact all the teachers who are responsible for the stalls on the C-rugby field with an expected number of 100 or more guests. Complete the code for the **Option B** button by formulating an SQL statement to display the **names of all the teachers** who are responsible for these stalls.

Example of the output:

Teacher
Ferreira, G
Fouche, JC

(4)

- 1.3 Mr Sweety needs to calculate the estimated number of servings to prepare for each stall using a figure of 1,25 servings per expected guest. Complete the code for the **Option C** button by formulating an SQL statement to display the **name of the stall**, the **number of guests** expected and the **number of servings to prepare** for each stall in the **tblStalls** table. The **number of servings to prepare** is a calculated field which needs to be rounded off to the nearest whole number. Use **ServingsToPrepare** as the field name.

Example of the output for the first five records:

StallName	NumOfGuests	ServingsToPrepare
Gaming Palace	142	178
Blockbusters	12	15
Green Hills	110	138
Twenty Something	88	110
Blue Pilots	56	70

:

(5)



- 1.4 Complete the code for the **Option D** button by formulating an SQL statement to display the total amount of donations promised, but not yet received, for a particular item. Allow the user to enter the name of the item from the keyboard. Use **Total** as the calculated field name.

Example of the output, if **Rice** is the item entered by the user:

Total
R 5,135.00

(5)

- 1.5 Mr Du Plessis (the teacher for Grade 12B) wants to know what the total amount received is, in terms of donations, for each of the various items for the Grade 12B class. Complete the code for the **Option E** button by formulating an SQL statement to display the total amount received for each item for the stall operated by Grade 12B. Use **AmountReceived** as the new field name.

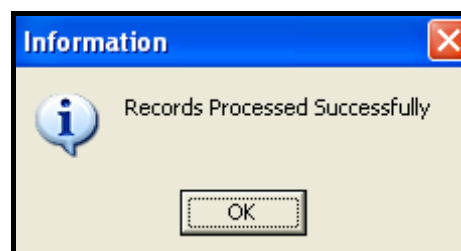
Example of the output:

Item	AmountReceived
Meat	R 100.00
Other expenses	R 105.00
Rice	R 195.00
Vegetables	R 150.00

(7)

- 1.6 Mr Sweety has been informed that the estimated number of guests for all the A-classes (that is Grades 8A, 9A, 10A, 11A and 12A) has increased by 5% according to the ticket sales. Complete the code for the **Option F** button by formulating an SQL statement that will **update** the **tblDonations** table accordingly.

Example of the output:



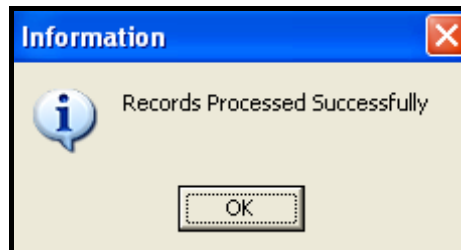
HINT: Select **Option A** to verify that the records have been updated.

(5)



- 1.7 One of the parents of a Grade 12B learner donated R200,00 towards 'other expenses' to the class. Grade 12B operates the stall with ID number HC77 (**StallID**). Complete the code for the **Option G** button by formulating an SQL statement that will add the donation to the records.

Example of the output:



HINT: Select **Option E** to verify that the record has been updated.

(4)

- Enter your examination number as a comment in the first line of the file named **Question1_UXXXX.pas** containing the SQL statements.
- Save the unit **Question1_UXXXX** and the project **Question1_PXXXX** (File/Save All).
- A printout of the code for the **Question1_UXXXX.pas** file will possibly be required (see Instruction 10 on page 3).

[33]

QUESTION 2: DELPHI OBJECT-ORIENTED PROGRAMMING

The Grade 11 learners have to raise money to host the Grade 12 farewell function at the end of the year. Software has been partially developed to assist with managing the income and expenses of the different events held to raise funds.

The given program in the **Question2_Delphi** folder consists of a class unit to describe a fundraising-event object and a main unit to create an array of fundraising-event objects. The program uses the array of objects to display specific required information.

The program is designed to cater for storing the name of an event, the name of the person who organised the event, the date when the event took place, as well as the income received and expenses incurred for an event.

Do the following:

- Rename the folder **Question2_Delphi** as **Question2_X** (where X must be replaced with your examination number).
- Open Delphi and then open the file **Question2_P.dpr** in the folder **Question2_X**.
- Go to 'File/Save As ...' and save the unit as **Question2_UXXXX** (where XXXX must be replaced with the last FOUR digits of your examination number).
- Open the unit **uEvent.pas**.
- Go to 'File/Save As ...' and save the unit as **uEventXXXX.pas** (where XXXX must be replaced with the last FOUR digits of your examination number).
- Go to 'File/Save Project As ...' and save the project as **Question2_PXXXX** (where XXXX must be replaced with the last FOUR digits of your examination number).

- 2.1 The class unit named **uEventXXXX.pas** contains a **TEvent** class that describes a single fundraising event.

Note the following:

- **Use the names of the fields indicated in QUESTION 2.1.1 in this class.**
- None of the fields in this class should be accessible to the main unit (form unit).
- All the methods of this class should be accessible to the main unit (form unit).
- Code should **not be repeated** to solve a given problem. Where possible, the given methods **must** be used to do tasks required to solve problems.
- Some of the code in the given class has been inserted as comments in order to get the unit to compile. You will be required to remove the comment symbols, as indicated in the questions that follow.
- In addition to modifying the given methods, you will be required to add code for new methods, as described in the questions that follow.

You are required to correct and complete the given code in the **uEventXXXX** unit by doing the following:

2.1.1 The constructor receives the following information of an event as parameters:

- **Name**, which refers to the name of the event
- **Person**, which refers to the name of the person who organised the event
- **Date** on which the event took place
- **Income** that was generated
- **Expenses** that had to be incurred

The class variables have been initialised in the constructor using the parameter values. However, the class variables have not been declared yet.

Do the following:

- Declare the class variables of the class using the variables' names, as used in the constructor.
- Remove the comment symbols from the statements initialising the class variables in the constructor.
- Remove the comment symbols from the return statement in the given **toString** method.

(5)

2.1.2 Write a method called **calculateProfit** to calculate and return the profit that was made at an event.

(3)

2.1.3 Write a method called **findTerm** that will determine and return a value (1, 2, 3 or 0) indicating the term during which the event took place.

The following applies:

- Period of term 1: January, February and March
- Period of term 2: April, May and June
- Period of term 3: July, August and September
- Period of term 4: October, November and December
No events were hosted during term 4.
- A value of 0 must be returned in the cases where a value other than 1, 2 or 3 was generated.

(7)

2.1.4 Write a method called **constructNameString** that will construct and return a new string containing the name of the organiser of an event in the following format:

Initials<space>Last name

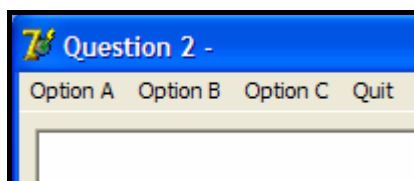
Example:

If the name of the person is Mr Johnson, John Patrick, the new string must be **JP Johnson**.

(10)



- 2.2 In the **Question2_UXXXX** unit (the main unit) you have been given code to display the following menu when you execute the program:



Open the **Question2_UXXXX** unit (the main unit).

Add your examination number as a comment in the first line of the **Question2_UXXXX** unit.

- 2.2.1 An array to store a maximum of twenty **TEvent**-type objects has been declared.

A text file named **Data2011.txt** containing information on all the fundraising events that took place during the year has been supplied in the **Question2_X** folder. The information on each event has been saved over three lines in the following format in the text file:

- **First line:** The name of the event followed by a colon (:) and the name of the organiser followed by the date of the event. Note that the name of the organiser and the date of the event are separated by the word 'on'.
- **Second line:** The income as an amount
- **Third line:** The expenses as an amount

Example of the content of the text file:

```
Fun Walk:Mr Jackson,Harold James on 2011/05/12
2500
500
Raffle#1:Mrs Freeman,Jane on 2011/02/24
545
0
Bazaar:Mrs Green,Lee-Ann Suzanne on 2011/04/26
780
300
Raffle#2:Mr Nthumba,Eric on 2011/03/11
375
0
LAN Gaming Competition:Mrs Xaba,Mary Elizabeth on 2011/08/01
2400
600
Raffle#3:Mr Wilson,Jeremy Peter Wayne on 2011/07/25
425
0
Pancake Sale:Mrs Moodley,Harriot Charlene on 2011/06/02
565
150
Battle of the Bands:Mr Anderson,Brandon Westley on 2011/02/18
8900
3000
Talent Competition:Mrs Mohammed,Judith Diane Sheryl on 2011/07/15
9500
2000
```

NOTE: The program must be able to read the information of an unknown number of events from the text file.

Write code in the **OnActivate event handler** of the form to read the information from the text file **Data2011.txt** as follows:

Test if the text file exists. Display a suitable message if the file does not exist and terminate the program.

If the file does exist, use a loop to read the information from the text file according to the following steps:

- Read the first line from the file and separate the text into the name of the event, the name of the organiser and the date of the event.
- Read the next two lines from the text file as the income and the expenses of the event.
- Use this information to create a new **TEvent** object and place the object into the array.
- Use a counter to keep track of how many objects have been placed in the array.

(10)

2.2.2 Menu Option A

When the user clicks on this option, the program must display all the information on all the events stored in the array.

Complete the code to display the output as follows:

Name	Organiser	Date	Income	Expenses
Fun Walk	Mr Jackson, Harold James	2011/05/12	R 2,500.00	R 500.00
Raffle#1	Mrs Freeman, Jane	2011/02/24	R 545.00	R 0.00
Bazaar	Mrs Green, Lee-Ann Suzanne	2011/04/26	R 780.00	R 300.00
Raffle#2	Mr Nthumba, Eric	2011/03/11	R 375.00	R 0.00
LAN Gaming Competition	Mrs Xaba, Mary Elizabeth	2011/08/01	R 2,400.00	R 600.00
Raffle#3	Mr Wilson, Jeremy Peter Wayne	2011/07/25	R 425.00	R 0.00
Pancake Sale	Mrs Moodley, Harriot Charlene	2011/06/02	R 565.00	R 150.00
Battle of the Bands	Mr Anderson, Brandon Westley	2011/02/18	R 8,900.00	R 3,000.00
Talent Competition	Mrs Mohammed, Judith Diane Sheryl	2011/07/15	R 9,500.00	R 2,000.00

(2)

2.2.3 Menu Option B

When the user clicks on this option, the program must display a heading and call the relevant methods to display the name of the organiser and the profit that was made during each event. The name of the organiser must be displayed in the following format:

Initials<space>Last name

Example of the output (on the next page):



Organiser	Profit
HJ Jackson	R 2,000.00
J Freeman	R 545.00
LS Green	R 480.00
E Nthumba	R 375.00
ME Xaba	R 1,800.00
JPW Wilson	R 425.00
HC Moodley	R 415.00
BW Anderson	R 5,900.00
JDS Mohammed	R 7,500.00

(4)

2.2.4 Menu Option C

When the user clicks on this option, the program must use the **findTerm** and the **calculateProfit** methods to determine and display the total profit that was made during each term.

Example of the output:

Profit per term		
Term 1	Term 2	Term 3
R 6,820.00	R 2,895.00	R 9,725.00

(8)

- Make sure your examination number is entered as a comment in the first line of the main unit **Question2_UXXXX.pas**, as well as the class unit **uEventXXXX.pas**.
- Save all the files (File/Save All).
- Printouts of the code for the units **Question2_UXXXX.pas** and **uEventXXXX.pas** may be required (see Instruction 10 on page 3).

[49]



QUESTION 3: DELPHI PROGRAMMING

A treasure hunt is one of the activities used as a fund-raiser during an annual market-day event at the school.

Participants will be required to pay an entry fee and select five different numbers from 1 to 20. Points are awarded for each number selected. Depending on the total number of points a participant has, different prizes are awarded.

Rules of the game:

- The numbers from 1 to 20 are used in the treasure hunt for participants to select from.
- A number of points are hidden behind each number.
- The participant must enter five different numbers in the range of 1 to 20.
- The hidden number of points awarded to each number that was entered will be added together to determine the total number of points for the participant.
- The participant is not allowed to enter the same number more than once. Every time a previously entered number is selected again, five points will be deducted from the total number of points for the participant.
- Every time a number outside the range of 1 to 20 is entered, five points will be deducted from the total number of points for the participant.
- A prize will be awarded to the participant depending on the total number of points that was scored.

NOTE: This is a problem-solving question. The following applies:

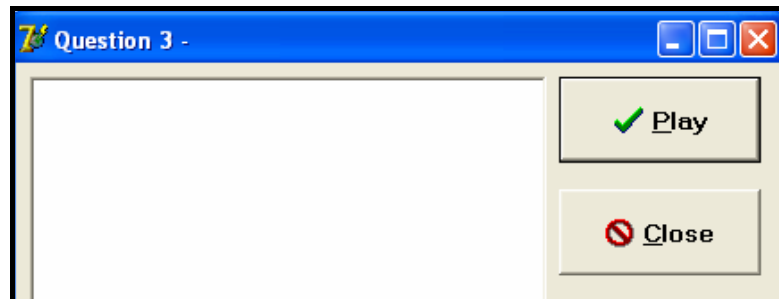
- You have to develop your own solution according to the specifications in the question paper.
- Good programming principles should be followed, for example descriptive variable names, indentation, et cetera.
- A modular programming approach should be followed. A maximum of **four marks** could be deducted if your solution is not modular, that is it has no subprograms with parameter passing.
- The program only has to work for ONE participant.

You have been supplied with an incomplete program in the **Question3_ Delphi** folder.

Do the following:

- Rename the folder named **Question3_Delphi** as **Question3_X** (where X should be replaced with your examination number).
- Open Delphi and then open the file called **Question3_P.dpr** in the folder **Question3_X**.
- Go to 'File/Save As ...' and save the unit as **Question3_UXXXX** (where XXXX should be replaced with the last FOUR digits of your examination number).
- Go to 'File/Save Project As ...' and save the project as **Question3_PXXXX** (where XXXX should be replaced with the last FOUR digits of your examination number).
- Add your examination number to the right of the caption of the form.
- Execute the program. The following interface will be displayed (on the next page):





A list of numbers from 1 to 20 and their corresponding points are stored in an array called **arrPoints**.

NOTE: The 20 elements in the array are not in numeric order.

Example of the declaration of the array **arrPoints**:

```
arrPoints : Array[1..20] of string =
    ( '12:40', '20:0', '13:0', '3:0', '15:0', '9:0', '19:50', '10:0',
      '8:90', '11:0', '1:0', '5:30', '16:0', '14:100', '4:0', '17:0',
      '18:20', '6:0', '7:0', '2:20' );
```

The format of each entry in the array is as follows:

'12:40': **12** refers to the number (that the participant will select) and **40** refers to the number of points hidden behind number 12.

'20:0': **20** refers to the number (that the participant will select) and **0** refers to the number of points hidden behind number 20, et cetera.

Complete the code for the **Play** button to do the following:

3.1 Allow the participant to enter five numbers.

Create an array to store the entries entered by the user.

The following must be done for each number entered:

- Add the corresponding points for the number entered to the participant's total.
- Delete the entry from the given array. Example: If 8 is entered, the entry '8:90' must be deleted from array arrPoints.
- Store the entry in a new array in the same format as it is in the given array. Example: If 8 is entered, the entry '8:90' must be stored in the new array.
- If a number is entered more than once:
 - Five points must be deducted from the participant's total points each time.
 - The entry is stored as 'number ALREADY SELECTED'. Example: If 8 is entered a second time, the entry in the new array will be '8 ALREADY SELECTED'.



- If a number outside the range of 1 to 20 is entered:
 - Five points must be deducted from the participant's total points each time.
 - The entry is stored as 'number INVALID NUMBER' in the new array.
Example: If 21 is entered, the entry in the new array will be '21 INVALID NUMBER'.

(26)

3.2 Display the following:

- The remaining entries in the treasure hunt after the participant's choices
- The participant's choices, the points achieved and the corresponding prize

The prize awarded to the participant will depend on the number of total points scored.

Total Points	Prize
≤ 0	No Prize
1–100	Teddy Bear
101–200	Fishing Rod
> 200	Gym Membership

(12)

Example test runs:

Example 1:

Input:

The following numbers are entered one by one: **2; 5; 19; 14; 6.**

Output (on the next page):




```
Numbers not selected
=====
12:40
20:0
13:0
3:0
15:0
9:0
10:0
8:90
11:0
1:0
16:0
4:0
17:0
18:20
7:0

Participant's choices
=====
2:20
5:30
19:50
14:100
6:0
Points: 200
Prize: Fishing Rod
```

Example 2:

Input:

The following numbers are entered one by one: **2; 3; 4; 45; 2.**

Output (on the next page):



```
Numbers not selected
=====
12:40
20:0
13:0
15:0
9:0
19:50
10:0
8:90
11:0
1:0
5:30
16:0
14:100
17:0
18:20
6:0
7:0

Participant's choices
=====
2:20
3:0
4:0
45 INVALID NUMBER
2 ALREADY SELECTED
Points: 10
Prize: Teddy Bear
```

- Make sure your examination number is entered as a comment in the first line of the main unit **Question3_UXXXX**.
- Save all the files (File/Save All).
- A printout of the code for the unit **Question3_UXXXX** may be required (see Instruction 10 on page 3).

[38]**TOTAL SECTION A: 120**

SECTION B

Answer ALL the questions in this section only if you studied **Java**.

SCENARIO

Fund-raising has become an integral part of an annual school programme. The community, together with the teachers and learners at Sundowns High School, put their efforts together in organising several events throughout the year to ensure that the school has sufficient funds to support the school's academic, sporting and cultural activities.

QUESTION 1: JAVA PROGRAMMING AND DATABASE

One fundraising event the school decided to have is a curry and rice day. Mr Sweetey, the fundraising coordinator at the school, has created a database named **FundsDB** to store the data for the curry and rice day.

Each class is responsible for a stall that will be set up, either on rugby fields A, B or C or on hockey fields A, B or C. Space is allocated to each stall according to the estimated number of guests that will visit the stall. Donations are collected towards the purchasing of some of the requirements (meat, rice, vegetables or other items) that are needed to prepare and serve the curry and rice.

An incomplete program has been developed to process queries on the data in the given database. Your task will be to complete this program.

The database named **FundsDB**, as well as an incomplete Java program, has been saved in the folder named **Question1_Java**. The folder contains a test class named **TestQuestion1.java** and an object class named **Funds.java** which will display the results of the queries.

NOTE: The design of the tables in the **FundsDB** database and the sample data for this question can be found in **ADDENDUM A: Table Description Sheet**.

NOTE: If you cannot use the database provided, follow the instructions in **ADDENDUM B** to create the database before you answer any of QUESTIONS 1.1 to 1.7.

NOTE: Make a copy of the **FundsDB** database BEFORE you start with the solution. You will need a copy of the original database to be able to test your program thoroughly.

Do the following:

- Rename the folder **Question1_Java** as **Question1_X**, where X should be replaced with your examination number.
- Open the incomplete program **TestQuestion1.java** in the **Question1_X** folder.
- Change the name of the class to **TestQuestion1XXXX** (where XXXX should be replaced with the last FOUR digits of your examination number). Save the file.
- Save the class as **TestQuestion1XXXX.java** (where XXXX should be replaced with the last FOUR digits of your examination number).

The connectivity code as well as the code to display the results has already been written as part of the given code in the file named **Funds.java**.

NOTE: If your program cannot connect to the database, make sure that the database file **FundsDB** is in the same folder as your program. If this is not the case, copy the database file **FundsDB** into the same folder as your program. Your program will not work if the database file is in a folder other than the folder containing your program.

NOTE: If you still cannot establish connectivity with the database when you execute the program, you must still do the SQL code and submit it for marking.

Marks will only be awarded for the programming code which contains the SQL statements in the file named TestQuestion1XXXX.java.

When you compile and execute the **TestQuestion1XXXX.java** file, the menu below will be displayed. However, if you enter any of the options (A to G), the program will not work because of the incomplete SQL statements.

MENU

Option A

Option B

Option C

Option D

Option E

Option F

Option G

Q - QUIT

Your Choice?

Do the following:

Complete the SQL statements in the **TestQuestion1XXXX.java** file for each menu option, as indicated in QUESTIONS 1.1 to 1.7 below. The code to pass the SQL statements to the relevant methods in the **Funds.java** file has been given to you. You need only complete the SQL statements and some input statements as required in the **TestQuestion1XXXX.java** file.

- 1.1 Mr Sweety wants a list of the estimated number of guests that will visit the stalls. Complete the code for **Option A** by formulating an SQL statement to display the **stall name**, **class** and **number of guests** for all the stalls in the **tblStalls** table. Display the results according to the number of guests in descending order.

Example of the output for the first five records:

StallName	Class	NumOfGuests
Crazy Dolls	9B	144
Gaming Palace	8A	142
The Rock	11A	136
The WW	9C	126
Double Espresso	9E	122

:

(3)

- 1.2 Miss Shelby, the deputy principal, needs to contact all the teachers who are responsible for the stalls on the C-rugby field with an expected number of 100 or more guests. Complete the code for **Option B** by formulating an SQL statement to display the **names of all the teachers** who are responsible for these stalls.

Example of the output:

Teacher
Ferreira, G
Fouche, JC

(4)

- 1.3 Mr Sweety needs to calculate the estimated number of servings to prepare for each stall using a figure of 1,25 servings per expected guest. Complete the code for **Option C** by formulating an SQL statement to display the **name of the stall**, the **number of guests** expected and the **number of servings to prepare** for each stall in the **tblStalls** table. The **number of servings to prepare** is a calculated field which needs to be rounded off to the nearest whole number. Use **ServingsToPrepare** as the field name.

Example of the output for the first five records:

StallName	NumOfGuests	ServingsToPrepare
Gaming Palace	142	178.0
Blockbusters	12	15.0
Green Hills	110	138.0
Twenty Something	88	110.0
Blue Pilots	56	70.0

:

(5)

- 1.4 Complete the code for **Option D** by formulating an SQL statement to display the total amount of donations promised, but not yet received, for a particular item. Allow the user to enter the name of the item from the keyboard. Use **Total** as the calculated field name.



Example of the output, if **Rice** is the item entered by the user:

Total
=====
R 5135.00

(5)

- 1.5 Mr Du Plessis (the teacher for Grade 12B) wants to know what the total amount received is, in terms of donations, for each of the various items for the Grade 12B class. Complete the code for **Option E** by formulating an SQL statement to display the total amount received for each item for the stall operated by Grade 12B. Use **AmountReceived** as the new field name.

Example of the output:

Item	AmountReceived
=====	
Meat	R 100.00
Other expenses	R 105.00
Rice	R 195.00
Vegetables	R 150.00

(7)

- 1.6 Mr Sweety has been informed that the estimated number of guests for all the A-classes (that is Grades 8A, 9A, 10A, 11A and 12A) has increased by 5% according to the ticket sales. Complete the code for **Option F** by formulating an SQL statement that will **update** the **tblDonations** table accordingly.

Example of the output:

Records Processed Successfully

HINT: Run **Option A** to verify that the records have been updated.

(5)

- 1.7 One of the parents of a Grade 12B learner donated R200,00 towards 'other expenses' to the class. Grade 12B operates the stall with ID number HC77 (**StallID**). Complete the code for **Option G** by formulating an SQL statement that will add the donation to the records.

Example of the output:

Records Processed Successfully

HINT: Run **Option E** to verify that the record has been updated.

(4)

- Enter your examination number as a comment in the first line of the file named **TestQuestion1XXXX.java** containing the SQL statements.
- Save the **TestQuestion1XXXX.java** file.
- A printout of the code for the **TestQuestion1XXXX.java** file will possibly be required (see Instruction 10 on page 3).

[33]

QUESTION 2: JAVA OBJECT-ORIENTED PROGRAMMING

The Grade 11 learners have to raise money to host the Grade 12 farewell function at the end of the year. Software has been partially developed to assist with managing the income and expenses of the different events held to raise funds.

The given program in the **Question2_Java** folder consists of an object class to describe a fundraising-event object and a test class to create an array of fundraising-event objects. The program uses the array of objects to display specific required information.

The program is designed to cater for storing the name of an event, the name of the person who organised the event, the date when the event took place, as well as the income received and expenses incurred for an event.

Do the following:

- Rename the folder **Question2_Java** as **Question2_X** (where X must be replaced with your examination number).
- Rename the **Event.java** file in the folder **Question2_X** as **EventXXXX.java** (where XXXX must be replaced with the last FOUR digits of your examination number).
- Open the **EventXXXX.java** file.
- Change the **class name** and the name of the **constructor method** to **EventXXXX** (where XXXX must be replaced with the last FOUR digits of your examination number).
- Add your examination number as a comment in the first line of the **EventXXXX.java** class. Save the file.
- Rename the **TestQuestion2.java** file in the folder **Question2_X** as **TestQuestion2XXXX.java** (where XXXX must be replaced with the last FOUR digits of your examination number).
- Open the **TestQuestion2XXXX.java** file.
- Change the **class name** to **TestQuestion2XXXX** (where XXXX must be replaced with the last FOUR digits of your examination number). Save the file.

- 2.1 The object class named **EventXXXX.java** will describe a single fundraising event.

Note the following:

- **Use the names of the fields indicated in QUESTION 2.1.1 in this class.**
- None of the fields in this class should be accessible to the test (application) class.
- All the methods of this class should be accessible to the test (application) class.
- Code should **not be repeated** to solve a given problem. Where possible, the given methods **must** be used to do tasks required to solve problems.
- Some of the code in the given class has been inserted as comments in order to get the class to compile. You will be required to remove the comment symbols, as indicated in the questions that follow.

- In addition to modifying the given methods, you will be required to add code for new methods, as described in the questions that follow.

You are required to correct and complete the given code in the **EventXXXX.java** class by doing the following:

2.1.1 The constructor receives the following information of an event as parameters:

- **Name**, which refers to the name of the event
- **Person**, which refers to the name of the person who organised the event
- **Date** on which the event took place
- **Income** that was generated
- **Expenses** that had to be incurred

The class variables have been initialised in the constructor using the parameter values. However, the class variables have not been declared yet.

Do the following:

- Declare the class variables of the class using the variables' names, as used in the constructor.
- Remove the comment symbols from the statements initialising the class variables in the constructor.
- Remove the comment symbols from the return statement in the given **toString()** method.

(5)

2.1.2 Write a method called **calculateProfit()** to calculate and return the profit that was made at an event.

(3)

2.1.3 Write a method called **findTerm()** that will determine and return a value (1, 2, 3 or 0) indicating the term during which the event took place.

The following applies:

- Period of term 1: January, February and March
- Period of term 2: April, May and June
- Period of term 3: July, August and September
- Period of term 4: October, November and December
- No events were hosted during term 4.
- A value of 0 must be returned in the cases where a value other than 1, 2 or 3 was generated.

(7)

- 2.1.4 Write a method called **constructNameString()** that will construct and return a new string containing the name of the organiser of an event in the following format:

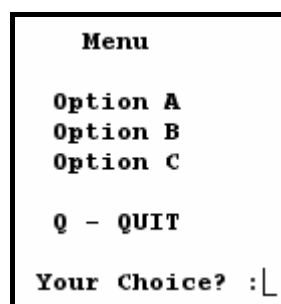
Initials<space>Last name

Example:

If the name of the person is Mr Johnson, John Patrick, the new string must be **JP Johnson**.

(10)

- 2.2 In the **TestQuestion2XXXX.java** file (the test class) you have been given code to display the following menu when you execute the program:



Open the **TestQuestion2XXXX.java** file (the test class).

Add your examination number as a comment in the first line of the **TestQuestion2XXXX.java** class.

- 2.2.1 An array to save a maximum of twenty **Event**-type object has been declared.

NOTE: Replace XXXX in the declaration statement of the array with the last four digits of your examination number.

A text file named **Data2011.txt** containing information on all the fundraising events that took place during the year has been supplied in the **Question2_X** folder. The information on each event has been saved over three lines in the following format in the text file:

- **First line:** The name of the event followed by a colon (:) and the name of the organiser followed by the date of the event. Note that the name of the organiser and the date of the event are separated by the word 'on'.
- **Second line:** The income as an amount
- **Third line:** The expenses as an amount

Example of the content of the text file (on the next page):

```
Funwalk:Mr Jackson,Harold James on 2011/05/12
2500
500
Raffle#1:Mrs Freeman,Jane on 2011/02/24
545
0
Food Bazaar:Mrs Green,Lee-Ann Suzanne on 2011/04/26
780
300
Raffle#2:Mr Nthumba,Eric on 2011/03/11
375
0
LAN Gaming Competition:Mrs Xaba,Mary Elizabeth on 2011/08/01
2400
600
Raffle#3:Mr Wilson,Jeremy Peter Wayne on 2011/07/25
425
0
Pancake sale:Mrs Moodley,Harriot Charlene on 2011/06/02
565
150
Battle of the Bands:Mr Anderson,Brandon Westley on 2011/02/18
8900
3000
Talent Competition:Mrs Mohammed,Judith Diane Sheryl on 2011/07/15
9500
2000
```

NOTE: The program must be able to read the information of an unknown number of events from the text file.

Write code to read the information from the text file **Data2011.txt** as follows:

Test if the text file exists. Display a suitable message if the file does not exist and terminate the program.

If the file does exist, use a loop to read the information from the text file according to the following steps:

- Read the first line from the file and separate the text into the name of the event, the name of the organiser and the date of the event.
- Read the next two lines from the text file as the income and the expenses of the event.
- Use this information to create a new **EventXXXX** object and place the object into the array.
- Use a counter to keep track of how many objects have been placed in the array.

(10)

2.2.2 Menu Option A

When the user selects this menu option, the program must display all the information on all the events stored in the array.

Complete the code to display the output as follows (on the next page):

Name	Organiser	Date	Income	Expenses
Fun Walk	Mr Jackson, Harold James	2011/05/12	R 2500.00	R 500.00
Raffle#1	Mrs Freeman, Jane	2011/02/24	R 545.00	R 0.00
Bazaar	Mrs Green, Lee-Ann Suzanne	2011/04/26	R 780.00	R 300.00
Raffle#2	Mr Nthumba, Eric	2011/03/11	R 375.00	R 0.00
LAN Gaming Competition	Mrs Xaba, Mary Elizabeth	2011/08/01	R 2400.00	R 600.00
Raffle#3	Mr Wilson, Jeremy Peter Wayne	2011/07/25	R 425.00	R 0.00
Pancake Sale	Mrs Moodley, Harriot Charlene	2011/06/02	R 565.00	R 150.00
Battle of the Bands	Mr Anderson, Brandon Westley	2011/02/18	R 8900.00	R 3000.00
Talent Competition	Mrs Mohammed, Judith Diane Sheryl	2011/07/15	R 9500.00	R 2000.00

(2)

2.2.3 Menu Option B

When the user selects this menu option, the program must display a heading and call the relevant methods to display the name of the organiser and the profit that was made during each event. The name of the organiser must be displayed in the following format:

Initials<space>Last name

Example of the output:

Organiser	Profit
HJ Jackson	R 2000.00
J Freeman	R 545.00
LS Green	R 480.00
E Nthumba	R 375.00
ME Xaba	R 1800.00
JPW Wilson	R 425.00
HC Moodley	R 415.00
BW Anderson	R 5900.00
JDS Mohammed	R 7500.00

(4)

2.2.4 Menu Option C

When the user selects this menu option, the program must use the **findTerm()** and the **calculateProfit()** methods to determine and display the total profit that was made during each term.

Example of the output:

Profit per term		
Term 1	Term 2	Term 3
R 6820.00	R 2895.00	R 9725.00

(8)

- Make sure your examination number is entered as a comment in the first line of the test class **TestQuestion2XXXX.java**, as well as the object class **EventXXXX.java**.
- Save all the files (File/Save All).
- Printouts of the code for the classes **TestQuestion2XXXX.java** and **EventXXXX.java** will possibly be required (see Instruction 10 on page 3).

[49]



QUESTION 3: JAVA PROGRAMMING

A treasure hunt is one of the activities used as a fund-raiser during an annual market-day event at the school.

Participants will be required to pay an entry fee and select five different numbers from 1 to 20. Points are awarded for each number selected. Depending on the total number of points a participant has, different prizes are awarded.

Rules of the game:

- The numbers from 1 to 20 are used in the treasure hunt for participants to select from.
- A number of points are hidden behind each number.
- The participant must enter five different numbers in the range of 1 to 20.
- The hidden number of points awarded to each number that was entered will be added together to determine the total number of points for the participant.
- The participant is not allowed to enter the same number more than once. Every time a previously entered number is selected again, five points will be deducted from the total number of points for the participant.
- Every time a number outside the range of 1 to 20 is entered, five points will be deducted from the total number of points for the participant.
- A prize will be awarded to the participant depending on the total number of points that was scored.

NOTE: This is a problem-solving question. The following applies:

- You have to develop your own solution according to the specifications in the question paper.
- Good programming principles should be followed, for example descriptive variable names, indentation, et cetera.
- A modular programming approach should be followed. A maximum of **four marks** could be deducted if your solution is not modular, that is it has no subprograms with parameter passing.
- You may use one or more classes for this solution, but the program only has to work for ONE participant.

You have been given an incomplete program in the **Question3_Java** folder.

Do the following:

- Rename the folder named **Question3_Java** as **Question3_X** (where X should be replaced with your examination number).
- Rename the file **TestQuestion3.java** in this folder as **TestQuestion3XXXX.java** (where XXXX should be replaced with the last FOUR digits of your examination number).
- Open the file (incomplete program) **TestQuestion3XXXX.java**.
- Change the class name to **TestQuestion3XXXX** (where XXXX should be replaced with the last FOUR digits of your examination number).
- Add your examination number as a comment in the first line of the program.



A list of numbers from 1 to 20 and their corresponding points are stored in an array called **arrPoints**.

NOTE: The 20 elements in the array are not in numeric order.

Example of the declaration of the array **arrPoints**:

```
String [] arrPoints = { "12:40", "20:0", "13:0", "3:0", "15:0", "9:0",  
                        "19:50", "10:0", "8:90", "11:0", "1:0", "5:30",  
                        "16:0", "14:100", "4:0", "17:0", "18:20", "6:0",  
                        "7:0", "2:20" };
```

The format of each entry in the array is as follows:

'12:40': **12** refers to the number (that the participant will select) and **40** refers to the number of points hidden behind number 12.

'20:0': **20** refers to the number (that the participant will select) and **0** refers to the number of points hidden behind number 20, et cetera.

Complete the code to do the following:

3.1 Allow the participant to enter five numbers.

Create an array to store the entries entered by the user.

The following must be done for each number entered:

- Add the corresponding points for the number entered to the participant's total.
- Delete the entry from the given array. Example: If 8 is entered, the entry '8:90' must be deleted from array **arrPoints**.
- Store the entry in a **new array** in the same format as it is in the given array. Example: If 8 is entered, the entry '8:90' must be stored in the new array.
- If a number is entered more than once:
 - Five points must be deducted from the participant's total points each time.
 - The entry is stored as 'number ALREADY SELECTED'.
Example: If 8 is entered a second time, the entry in the new array will be '8 ALREADY SELECTED'.
- If a number outside the range of 1 to 20 is entered:
 - Five points must be deducted from the participant's total points each time.
 - The entry is stored as 'number INVALID NUMBER' in the new array.
Example: If 21 is entered, the entry in the new array will be '21 INVALID NUMBER'.

(26)



3.2 Display the following:

- The remaining entries in the treasure hunt after the participant's choices
- The participant's choices, the points achieved and the corresponding prize

The prize awarded to the participant will depend on the number of total points scored.

Total Points	Prize
≤ 0	No Prize
1–100	Teddy Bear
101–200	Fishing Rod
> 200	Gym Membership

(12)

Example test runs:**Example 1:**

Input:

The following numbers are entered one by one: **2; 5; 19; 14; 6.**

Output:

Numbers not selected
=====
12:40
20:0
13:0
3:0
15:0
9:0
10:0
8:90
11:0
1:0
16:0
4:0
17:0
18:20
7:0
Participant's choices
=====
2:20
5:30
19:50
14:100
6:0
Points: 200
Prize: Fishing Rod



Example 2:

Input:

The following numbers are entered one by one: 2; 3; 4; 45; 2.

Output:

```

Numbers not selected
=====
12:40
20:0
13:0
15:0
9:0
19:50
10:0
8:90
11:0
1:0
5:30
16:0
14:100
17:0
18:20
6:0
7:0

Participant's choices
=====
2:20
3:0
4:0
45 INVALID NUMBER
2 ALREADY SELECTED
Points: 10
Prize: Teddy Bear

```

- Make sure your examination number is entered as a comment in the first line of the class **TestQuestion3XXXX.java**, as well as any other class(es) you have created with code.
- Save the class(es).
- A printout of the code for the class **TestQuestion3XXXX.java**, as well as any other class(es) you have created, will possibly be required (see Instruction 10 on page 3).

[38]

TOTAL SECTION B: 120
GRAND TOTAL: 120



ADDENDUM A: Table Description Sheet

This sheet shows the data structure and sample data for the tables used in the **FundsDB** database in **Question 1**.

tblStalls Table Structure

tblStalls : Table			
Field Name	Data Type	Description	
StallID	Text	R=Rubby/H=Hockey; A/B/C indicates which field; unique two-digit number	
Class	Text	The class responsible for the administration of the stall	
StallName	Text	The name of the stall	
Teacher	Text	The teacher responsible for the stall	
NumOfGuests	Number	The expected number of guests	

tblStalls Table Sample Data

tblStalls : Table				
StallID	Class	StallName	Teacher	NumOfGuests
HA51	8B	Blue Pilots	Freulich, I	56
HA52	12F	The Big Boys	Jones, H	78
HA54	12A	Almost Eighteen	du Toit, ER	98
HA55	8D	Green Hills	Mendes, I	110
HB61	9C	The WW	van Wyk, VW	126
HB62	10B	Dumela	Nguni, S	88
HB63	10D	Old School	Bekker, L	54
HB64	8C	Twenty Something	Khoza, B	88
HC71	9A	Grade 9A	Baker, VWJ	48
HC72	12D	Mammas Boys	Baker, SS	80
HC73	9D	Kawazaki's Home Run	Kowalski, J	100
HC74	10A	Hip Hop	van der Merwe, L	16
HC75	8E	Blockbusters	Honeywell, L	12
HC76	10C	Yellow Polkadot	Smythe, K	20
HC77	12B	BornFree	du Plessis, LK	76
RA11	11D	Blue Berry Pie	Swarts, A	86
RA12	10E	The Jazz Club	Smith, KH	80

tblDonations Table Structure

tblDonations : Table				
Field Name	Data Type	Description		
DonationID	AutoNumber	A unique number for each donation		
StallID	Text	The ID of the stall for which the donation is intended		
Item	Text	The item towards which the donation was made		
Amount	Currency	The amount donated towards buying the item		
Received	Yes/No	Donation received? Yes=True, No=False		

tblDonations Table Sample Data

tblDonations : Table				
DonationID	StallID	Item	Amount	Received
1	RC34	Meat	R 50.00	False
2	HC75	Rice	R 65.00	False
4	HB64	Other expenses	R 35.00	True
5	HA51	Rice	R 65.00	False
6	HC73	Meat	R 50.00	True
8	RC32	Rice	R 65.00	True
9	HC71	Vegetables	R 75.00	False
10	RB24	Meat	R 50.00	False
11	HB62	Rice	R 65.00	False
13	RC31	Meat	R 50.00	False
14	HB63	Meat	R 50.00	True
15	HC74	Meat	R 50.00	False
16	RA12	Meat	R 50.00	True
17	RB23	Rice	R 65.00	True
18	RA11	Other expenses	R 35.00	True
19	RB21	Meat	R 50.00	False
20	RA13	Meat	R 50.00	True



ADDENDUM B: Instructions to create the database FundsDB.mdb

If you cannot use the database provided, do the following:

- Use the two text files named **tblStalls** and **tblDonations** that have been supplied. Create your own database with the name **FundsDB** that includes a table named **tblStalls** and another table named **tblDonations** in the folder called **Question1_Delphi** or **Question1_Java**.
- Change the data types and the sizes of the fields in the two tables according to the specifications given below.

The **tblStalls** table stores data on the stalls at the event. The fields in the **tblStalls** table are defined as follows:

<u>Field Name</u>	<u>Type</u>	<u>Size</u>	<u>Description</u>
StallID	Text	5	R=Rugby/H=Hockey; A/B/C indicates which field; unique two-digit number
Class	Text	3	The class responsible for the administration of the stall
StallName	Text	30	The name of the stall
Teacher	Text	30	The teacher responsible for the stall
NumOfGuests	Number	Integer	The expected number of guests

See ADDENDUM A for an example of the data in the **tblStalls** table.

The **tblDonations** table stores data on the donations. The fields in the **tblDonations** table are defined as follows:

<u>Field Name</u>	<u>Type</u>	<u>Size</u>	<u>Description</u>
DonationID	AutoNumber	LongInt	A unique number for each donation
StallID	Text	5	The ID of the stall for which the donation is intended
Item	Text	30	The item towards which the donation was made
Amount	Currency	Double	The amount donated towards buying the item
Received	Yes/No	True/False	Donation received? Yes=True, No=False

See ADDENDUM A for an example of the data in the **tblDonations** table.



ADDENDUM C: Instructions to connect to the database in Delphi

If you cannot use the database provided, do the following:

- Click on the ADOQuery component named **qryQOne**.
- 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 box.
- 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 **FundsDB** file. You will find this file in the **Question1_Delphi** folder. Once you have found it, select the **FundsDB** file and click on the Open button.
- Remove the user name Admin.
- Click on the Test Connection button.
- Click OK on each one of the open dialogue windows.



INFORMATION TECHNOLOGY P1**FEBRUARY/MARCH 2012****INFORMATION SHEET** *(to be completed by the candidate)***120**

NAME OF PROVINCE _____

CENTRE NUMBER _____

EXAMINATION NUMBER _____

WORKSTATION NUMBER _____

DATE OF EXAMINATION _____

Programming language used
(Mark appropriate box with a cross (X).)

Delphi

Java

FOLDER NAME _____

Enter the file name used for each answer and tick if saved.

Question number	Saved (tick✓)	Maximum mark	Mark achieved	Marker initial/code
1		33		
2		49		
3		38		
TOTAL		120		

Comment *(for official use only)*



INLIGTINGSTECHNOLOGIE V1

FEBRUARIE/MAART 2012

INLIGTINGSBLAD (om deur die kandidaat voltooi te word)

120

NAAM VAN PROVINSIE

SENTRUMNUMMER

EKSAMENNUMMER

NOMMER VAN WERKSTASIE

DATUM VAN EKSAMEN

Programmeringstaal wat gebruik is
(Merk die toepaslike blokkie met 'n kruisie (X).)

Delphi	Java
--------	------

NAAM VAN LÊRGIDS

Skryf die naam neer van die lêer wat gebruik is en merk dit af as dit gestoor is.

Vraagnummer	Gestoor (merk✓)	Maksimum punt	Punt behaal	Nasiener se voorletters/kode
1		35		
2		49		
3		36		
TOTAAL		120		

Kommentaar (slegs vir amptelike gebruik)



BYLAE C: Instruksies om met die databasis in Delphi te koppel

Indien jy nie die databasis wat voorsien word, kan gebruik nie, doen die volgende:

- Klik op die 'ADOQuery'-komponent met die naam **qryQEn**.
- Klik op die 'Ellipse'-knopie (drie kolletjies) regs van die 'ConnectionString'-eienskap in die Objekt-inspekteur ('Object Inspector').
- Klik op die 'Build'-knopie wat jou na die 'Data Link Properties'-dialoogvenster lei.
- Klik op die 'Provider tab' om die 'Provider tab sheet' oop te maak en kies 'Microsoft Jet 4.0 OLE DB Provider'. Klik op die 'Next'-knopie.
- Die 'Connection tab sheet' sal vertoon word. Die eerste opsie op die 'Connection tab sheet' voorsien 'n 'Ellipse'-knopie (drie kolletjies) wat jou toelaat om deur te blaai en die **FondseDB**-lêer te soek. Jy sal hierdie lêer in die **Vraag1_Delphi**-lêergids ('folder') kry. Sodra jy dit gekry het, klik op die **FondseDB**-lêer en klik dan op die 'Open'-knopie.
- Verwyder die gebruikersnaam ('user name') 'Admin'.
- Klik op die 'Test Connection'-knopie.
- Klik 'OK' op elkeen van die oop dialoogvensterfjies.



BYLAE B: Instruksies om die databasis FondseDB.mdb te skep

Indien jy nie die databasis wat voorsien is, kan gebruik nie, doen die volgende:

- Gebruik die twee teksleërs met die name **tblStalletjies** en **tblDonasies** wat voorsien is. Skep jou eie databasis met die naam **FondseDB** met 'n tabel met die naam **tblStalletjies** en 'n ander tabel met die naam **tblDonasies** in die **Vraag1_Delphi**- of die **Vraag1_Java**-leërgids insluit.
- Verander die datatipes en die groottes van die velde in die twee tabelle volgens die spesifikasies wat hieronder gegee word.

Die **tblStalletjies**-tabel stoor data oor die stalletjies by die geleentheid. Die velde in **tblStalletjies**-tabel is soos volg gedefinieer:

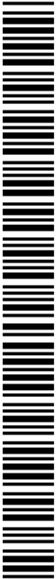
<u>Veldnaam</u>	<u>Tipe</u>	<u>Grootte</u>	<u>Beskrywing</u>
StalletjieID	Text	5	R=Rugby/H=Hokkie; A/B/C wat aandui watter veld; unieke tweekarakter-kode
Klas	Text	3	Die klas wat vir die administrasie van die stalletjie verantwoordelik is
StalletjieNaam	Text	30	Die naam van die stalletjie
Onderwyser	Text	30	Die onderwyser wat vir die stalletjie verantwoordelik is
GetalGaste	Number	Integer	Die getal gaste wat verwag word

Sien BYLAE A vir 'n voorbeeld van die data in die **tblStalletjies**-tabel.

Die **tblDonasies**-tabel stoor data oor die donasies. Die velde in die **tblDonasies**-tabel is soos volg gedefinieer:

<u>Veldnaam</u>	<u>Tipe</u>	<u>Grootte</u>	<u>Beskrywing</u>
DonasieID	AutoNumber	Longint	'n Unieke nommer vir elke donasie
StalletjieID	Text	5	Die ID van die stalletjie waarvoor die donasie bedoel is
Item	Text	30	Die item waarvoor die donasie gemaak is
Bedrag	Currency	Double	Die bedrag geskenk vir die aankoop van die item
Ontvang	Yes/No	True/False	Donasie ontvang? Ja=True, Nee=False

Sien BYLAE A vir 'n voorbeeld van die data in die **tblDonasies**-tabel.



BYLAAG A: Bladsy met beskrywings van tabelle

Hierdie bladsy toon die datastruktuur en voorbeelddata van die tabelle wat in die **FondseDB**-databasis in **Vraag 1** gebruik is.

tblStalletjies se Tabelstruktuur

tblStalletjies : Table				
Field Name	Data Type	Description		
StalletjieID	Text	R=Rugby/H=Hokkie; A/B/C wat aandui watter veld; unieke tweekarakter-kode		
Klas	Text	Die klas wat vir die administrasie van die stalletjie verantwoordelik is		
StalletjieNaam	Text	Die naam van die stalletjie		
Onderwyser	Text	Die onderwyser wat vir die stalletjie verantwoordelik is		
GetalGaste	Number	Die getal gaste wat verwaag word		

tblDonasies se Tabelstruktuur

tblDonasies : Table				
Field Name	Data Type	Description		
DonasieID	AutoNumber	'n Unieke nommer vir elke donasie		
StalletjieID	Text	Die ID van die stalletjie waarvoor die donasie bedoel is		
Item	Text	Die item waarvoor die donasie gemaak is		
Bedrag	Currency	Die bedrag geskenk vir die aankoop van die item		
Ontvang	Yes/No	Donasie ontvang? Ja=True, Nee=False		

tblStalletjies-tabel – Voorbeeld van Data

tblStalletjies : Table			
StalletjieID	Klas	StalletjieNaam	Onderwyser
HA51	8B	Blue Pilots	Freulich, I
HA52	12F	The Big Boys	Jones, H
HA54	12A	Almost Eighteen	du Toit, ER
HA55	8D	Green Hills	Mendes, I
HB61	9C	The WWV	van Wyk, W
HB62	10B	Dumela	Nguni, S
HB63	10D	Old School	Bekker, L
HB64	8C	Twenty Something	Khoza, B
HC71	9A	Grade 9A	Baker, WJ
HC72	12D	Mammass Boys	Baker, SS
HC73	9D	Kawazaki's Home Run	Kowalski, J
HC74	10A	Hip Hop	van der Merwe, L
HC75	8E	Blockbusters	Honeywell, L
HC76	10C	Yellow Polkadot	Smythe, K
HC77	12B	BornFree	du Plessis, LK
RA11	11D	Blue Berry Pie	Swarts, A
RA12	10E	The Jazz Club	Smith, KH

tblDonasies-tabel – Voorbeeld van Data

tblDonasies : Table				
DonasieID	StalletjieID	Item	Bedrag	Ontvang
1	RC34	Vleis	R 50.00	False
2	HC75	Rys	R 65.00	False
4	HB64	Ander uitgawes	R 35.00	True
5	HA51	Rys	R 65.00	False
6	HC73	Vleis	R 50.00	True
8	RC32	Rys	R 65.00	True
9	HC71	Groente	R 75.00	False
10	RB24	Vleis	R 50.00	False
11	HB62	Rys	R 65.00	False
13	RC31	Vleis	R 50.00	False
14	HB63	Vleis	R 50.00	True
15	HC74	Vleis	R 50.00	False
16	RA12	Vleis	R 50.00	True
17	RB23	Rys	R 65.00	True
18	RA11	Ander uitgawes	R 35.00	True
19	RB21	Vleis	R 50.00	False
20	RA13	Vleis	R 50.00	True



Voorbeeld 2:
Toevoer:

Die volgende getalle word een vir een ingesleutel: 2; 3; 4; 45; 2.

Afvoer:

=====	
Getalle nie gekies nie	
12:40	
20:0	
13:0	
15:0	
9:0	
19:50	
10:0	
8:90	
11:0	
1:0	
5:30	
16:0	
14:100	
17:0	
18:20	
6:0	
7:0	
=====	
Deelnemer se keuses	
2:20	
3:0	
4:0	
45 ONGELDIGE GETAL	
2 ALREEDS GEKIES	
Punte: 10	
Prys: Teddiebeer	

- Maak seker dat jou eksamennummer as kommentaar in die eerste reël van die klas **ToetsVraag3XXXX.java**, asook enige ander klas(se) wat jy met kode geskep het, ingetik is.
- Stoor die klas(se).
- 'n Drukstuk van die kode van die klas **ToetsVraag3XXXX.java**, asook enige ander klas(se) wat jy geskep het, sal moontlik vereis word (sien instruksie 10 op bladsy 3).

[38]

TOTAAL AFDELING B: 120
GROOTTOTAAL: 120



3.2

Vertoon die volgende:

- Die oorblywende inskrywings in die skattejag na die deelnemer se keuses
- Die deelnemer se keuses, die punte behaal en die ooreenstemmende prys

Die prys wat aan die deelnemer toegeken word, sal van die totale getal punte wat behaal is, afhang.

Totale Punte	Prys
<= 0	Geen prys
1-100	Teddiebeer
101-200	Visstok
> 200	Gimnasiumlidmaatskap

(12)

Voorbeeldtoetslopie:

Voorbeeld 1:

Toevoer:

Die volgende getalle word een vir een ingesleutel: 2; 5; 19; 14; 6.

Afvoer:

Getalle nie gekies nie	=====
12:40	
20:0	
13:0	
3:0	
15:0	
9:0	
10:0	
8:90	
11:0	
1:0	
16:0	
4:0	
17:0	
18:20	
7:0	
Deelnemer se keuses	=====
2:20	
5:30	
19:50	
14:100	
6:0	
Punte: 200	
Prys: Visstok	



'n Lys van getalle van 1 tot 20 en hul ooreenstemmende punte word in 'n skikking met die naam **arrPunte** gestoor.

NOTA: Die 20 elemente in die skikking is nie in numeriese volgorde nie.

Voorbeeld van die verklaring van die skikking **arrPunte**:

```
String [] arrPunte = { "12:40", "20:0", "13:0", "3:0", "15:0", "9:0",  
"19:50", "10:0", "8:90", "11:0", "1:0", "5:30",  
"16:0", "14:100", "4:0", "17:0", "18:20", "6:0",  
"7:0", "2:20" };
```

Die formaat van elke inskrywing in die skikking is soos volg:

'12:40': **12** verwys na die getal (wat die deelnemer sal kies) en **40** verwys na die aantal punte wat agter die getal 12 weggesteek is.

'20:0': **20** verwys na die getal (wat die deelnemer sal kies) en **0** verwys na die aantal punte wat agter die getal 20 weggesteek is, ensovoorts.

Voltooi die kode om die volgende te doen:

3.1 Laat die deelnemer toe om vyf getalle in te sleutel.

Skep 'n skikking om die inskrywings wat deur die gebruiker ingesleutel is, te stoor.

Die volgende moet vir elke getal wat ingesleutel is, gedoen word:

- Tel die ooreenstemmende punte vir die getal wat ingesleutel is, by die totaal van die deelnemer.
- Verwyder die inskrywing uit die gegewe skikking. Voorbeeld: Indien 8 ingesleutel word, moet die inskrywing '8:90' in die nuwe skikking gestoor word.
- Indien 'n getal meer as een keer ingesleutel word:
 - Vyf punte moet elke keer van die totale punte van die deelnemer afgetrek word.
 - Die inskrywing word as 'getal ALREEDS GEKIES' gestoor.
- Voorbeeld: Indien 8 vir 'n tweede keer ingesleutel word, sal die inskrywing in die nuwe skikking '8 ALREEDS GEKIES' wees.
- Indien 'n getal wat buite die interval van 1 tot 20 val, ingesleutel word:
 - Vyf punte moet elke keer van die totale punte van die deelnemer afgetrek word.
 - Die inskrywing word as 'getal ONGELDIGE GETAL' in die nuwe skikking gestoor.
- Voorbeeld: Indien 21 ingesleutel word, sal die inskrywing in die nuwe skikking '21 ONGELDIGE GETAL' wees.

(26)



VRAAG 3: JAVA – PROGRAMMERING

'n Skattejag is een van die aktiwiteite wat as fondsinasamelingsprojek gebruik word tydens 'n jaarlikse markdaggeleentheid by die skool.

Daar word van deelnemers verwag om 'n inskrywingsfooi te betaal en vyf verskillende getalle van 1 tot 20 te kies. Punte word toegeken aan elke getal wat gekies is. Afhangend van die totale getal punte wat 'n deelnemer het, word verskillende pryse toegeken.

Reëls van die speljag:

- Die getalle 1 tot 20 word gebruik om uit te kies tydens die skattejag vir deelnemers.
- 'n Aantal punte is agter elke getal weggesteek.
- Die deelnemer moet vyf verskillende getalle in die interval van 1 tot 20 insleutel.
- Die weggesteelte aantal punte wat toegeken is aan elke getal wat ingesleutel is, sal opgetel word om die totale aantal punte van die deelnemer te bepaal.
- Die deelnemer word nie toegelaat om dieselfde getal meer as een keer in te sleutel nie. Elke keer wanneer 'n vorige ingesleutelde getal weer gekies word, word vyf punte van die totale getal punte van die deelnemer afgetrek.
- Elke keer wanneer 'n getal buite die interval van 1 tot 20 ingesleutel word, word vyf punte van die totale getal punte van die deelnemer afgetrek.
- 'n Prys word aan die deelnemer toegeken afhangend van die totale getal punte wat behaal is.

LET WEL: Hierdie is 'n probleemoplossingsvraag. Die volgende is van toepassing:

- Jy moet jou eie oplossing volgens die spesifikasies in die vraestel ontwikkel.
- Goë programmeringsbeginsels moet gevolg word, byvoorbeeld beskrywende veranderlikename, inkeping, ensovoorts.
- 'n Moduliëre programmeringsbenadering moet gevolg word. 'n Maksimum van **vier punte** kan afgetrek word indien jou oplossing nie moduliër is nie, dit wil sê dit het geen subprogramme met paramateroordrag nie.
- Jy mag een of meer klasse vir hierdie oplossing gebruik, maar die program hoef slegs vir EN deelnemer te werk.

'n Onvolledige program in die **Vraag3_Java**- lêergids ('folder') is aan jou gegee.

Doen die volgende:

- Hernoem die lêergids met die naam **Vraag3_Java** tot **Vraag3_X** (waar X met jou eksamenommer vervang moet word).
- Hernoem die lêer **ToetsVraag3.java** in hierdie lêergids tot **ToetsVraag3XXXX.java** (waar XXXX met die laaste VIER syfers van jou eksamenommer vervang moet word).
- Maak die lêer (onvolledige program) **ToetsVraag3XXXX.java** oop. Verander die naam van die klas na **ToetsVraag3XXXX** (waar XXXX met die laaste VIER syfers van jou eksamenommer vervang moet word).
- Voeg jou eksamenommer by as kommentaar in die eerste reël van die program.



2.2.3

Keuselys Opsie B

Naam	Organiseerder	Datum	Inkomstes	Uitgawes
Pretloep	Mrs. Jackson, Harold James	2011/05/12	R 2500.00	R 500.00
Uitloping#1	Mrs. Freeman, Jane	2011/02/24	R 545.00	R 0.00
Basaar	Mrs. Green, Lee-Anne Suzanne	2011/04/26	R 780.00	R 300.00
Uitloping#2	Mrs. Mthumba, Eric	2011/03/11	R 375.00	R 0.00
LAM-Speletjieskompetisie	Mrs. Kaba, Mary Elizabeth	2011/08/01	R 2400.00	R 600.00
Uitloping#3	Mrs. Wilson, Jeremy Peter Wayne	2011/07/25	R 425.00	R 0.00
Pannekoekverkoping	Mrs. Moodley, Harriot Charlene	2011/06/02	R 565.00	R 150.00
Orkoste in Aksie	Mrs. Anderson, Brandon Westley	2011/02/18	R 8900.00	R 3000.00
Talentkompetisie	Mrs. Mohammed, Judith Diane Sheryl	2011/07/15	R 9500.00	R 2000.00

Wanneer die gebruiker hierdie opsie op die keuselys kies, moet die program 'n opskrif vertoon en die relevante metodes roep om die naam van die organiseerder en die wins wat met elke geleentheid gemaak is, te vertoon. Die naam van die organiseerder moet in die volgende formaat vertoon word:

Voorletters<spasie>Van

Voorbeeld van die afvoer:

Organiseerder	Wins
HJ Jackson	R 2000.00
J Freeman	R 545.00
LS Green	R 480.00
E Mthumba	R 375.00
ME Kaba	R 1800.00
JPM Wilson	R 425.00
HC Moodley	R 415.00
BW Anderson	R 5900.00
JDS Mohammed	R 7500.00

(4)

2.2.4

Keuselys Opsie C

Wanneer die gebruiker hierdie opsie op die keuselys kies, moet die program die **vindKwartaal()**- en die **bereken Wins()**-metodes gebruik om die totale wins wat gedurende elke kwartaal gemaak is, te bepaal en te vertoon.

Voorbeeld van die afvoer:

Wins per kwartaal	Kwartaal 1	Kwartaal 2	Kwartaal 3
	R 6820.00	R 2895.00	R 9725.00

(8)

- Maak seker dat jou eksamennummer as kommentaar in die eerste reël van die toetsklas **ToetsVraag2XXXXX.java**, asook die objekklas **GeleentheidXXXXX.java**, ingetik is.
- Stoor al die lêers ('File/Save All').
- Drukstukke van die kode van die klasse **ToetsVraag2XXXXX.java** en **GeleentheidXXXXX.java** sal moontlik vereis word (sien Instruksie 10 op bladsy 3).

[49]



```

Pretloop:Mnr. Jackson, Harold James op 2011/05/12
500
Uitlotting#1:Mev. Freeman, Jane op 2011/02/24
545
Basaar:Mev. Green, Lee-Ann Suzanne op 2011/04/26
780
Uitlotting#2:Mnr. Nthumba, Eric op 2011/03/11
375
LAN-spel-etjieskompetisie:Mev. Xaba, Mary Elizabeth op 2011/08/01
2400
600
Uitlotting#3:Mnr. Wilson, Jeremy Peter Wayne op 2011/07/25
425
Pannekoekverkopning:Mev. Moodley, Harriot Charlene op 2011/06/02
565
150
Orkeste in Akste:Mnr. Anderson, Brandon Westley op 2011/02/18
8900
3000
Talentkompetisie:Mev. Mohammed, Judith Diane Sheryl op 2011/07/15
9500
2000

```

LET WEL: Die program moet die inligting van 'n onbekende aantal geleenthede uit die teksleër kan lees.

Skryf kode om die inligting soos volg uit die teksleër **Data2011A.txt** te lees:

Toets of die leër bestaan. Vertoon 'n gepaste boodskap indien die leër nie bestaan nie en beëindig die program.

Indien die leër bestaan, gebruik 'n lus om die inligting volgens die volgende stappe uit die teksleër te lees:

- Lees die eerste reël uit die leër en breek die teks op in die naam van die geleentheid, die naam van die organiseerder en die datum van die geleentheid.
- Lees die volgende twee reëls uit die teksleër as die inkomste en die uitgawes van die geleentheid.
- Gebruik hierdie inligting om 'n nuwe **GeleentheidXXXX**-objek te skep en plaas die objek in die skikking.
- Gebruik 'n teller om rekord te hou van hoeveel objekte in die skikking geplaas is.

(10)

2.2.2

Keuselys Opsie A

Wanneer die gebruiker hierdie opsie op die keuselys kies, moet die program alle inligting oor al die geleenthede wat in die skikking gestoor is, vertoon.

Vertoon die kode om die afvoer soos volg te vertoon (op die volgende bladsy):



2.1.4

Skryf 'n metode met die naam **skepNaamString()** wat 'n nuwe string sal saamstel en terugstuur wat die naam van die organiseerder van 'n geleentheid in die volgende formaat sal bevat:

Voorletters<spasie>Van

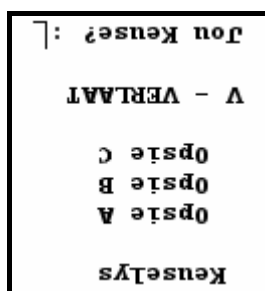
Voorbeeld:

Indien die naam van die persoon mnr. Johnson, John Patrick, is, moet die nuwe string **JP Johnson** wees.

(10)

2.2

In die **ToetsVraag2XXXXX.java**-lêer (die toetsklas) is kode aan jou gegee om die volgende keuselys te vertoon wanneer jy die program uitvoer:



Mak die **ToetsVraag2XXXXX.java**-lêer (die toetsklas) oop.

Voeg jou eksamennummer as kommentaar in die eerste reël van die **ToetsVraag2XXXXX.java**-klas by.

2.2.1

'n Skikking om 'n maksimum van twintig **Geleentheid**-tipe objekte te stoor, is verklaar.

NOTA: Vervang XXXX in die verklaringstelling van die skikking met die laaste vier syfers van jou eksamennummer.

'n Tekslêer met die naam **Data2011A.txt**, wat inligting bevat oor al die fondsinsamelingsgeleenthede wat gedurende die jaar plaasgevind het, is in die **Vraag2_X**-lêergids verskat. Die inligting oor elke geleentheid is oor drie reëls in die volgende formaat in die tekslêer gestoor:

- **Eerste reël:** Die naam van die geleentheid gevolg deur 'n dubbelpunt (:) en die naam van die organiseerder gevolg deur die datum waarop die geleentheid gehou is. Let daarop dat die naam van die organiseerder en die datum waarop die geleentheid gehou is, deur die woord 'op' geskei word.
- **Tweede reël:** Die inkomste as 'n bedrag
- **Derde reël:** Die uitgawes as 'n bedrag

Voorbeeld van die inhoud van die tekslêer (op die volgende bladsy):



- Bykomend tot die wysiging van die gegewe metodes sal daar van jou verwag word om kode vir nuwe metodes by te voeg, soos beskryf in die vrae wat volg.

Daar word van jou verwag om die gegewe kode in die **GelentheidXXXXX.java**-klas te korrigeer en te voltooi deur die volgende te doen:

2.1.1 Die konstruktur ('constructor') ontvang die volgende inligting oor 'n gelentheid as parameters:

- **Naam**, wat na die naam van die funksie verwys
- **Persoon**, wat na die naam van die persoon wat die funksie georganiseer het, verwys
- **Datum** waarop die funksie plaasgevind het
- **Inkomste** wat gegenereer is
- **Uitgawes** wat aangegaan is

Die klasveranderlikes is in die konstruktur geïnisiëleer deur die parameterwaardes te gebruik. Die klasveranderlikes is egter nog nie verklaar nie.

Doen die volgende:

- Verklaar die klasveranderlikes van die klas deur die veranderlikename wat in die konstruktur gebruik is, te gebruik.
- Verwyder die kommentaartekens uit die stellings wat die klasveranderlikes in die konstruktur inisiëleer.
- Verwyder die kommentaartekens uit die terugstuurstelling in die gegewe **toString()**-metode.

2.1.2 Skryf 'n metode met die naam **berekenWins()** om die wins wat met 'n gelentheid gemaak is, te bereken en terug te stuur. (3)

2.1.3 Skryf 'n metode met die naam **vindKwartaal()** wat 'n waarde (1, 2, 3 of 0), wat die kwartaal aandui waartydens die gelentheid plaasgevind het, sal bepaal en terugstuur.

Die volgende is van toepassing:

- Tydperk vir kwartaal 1: Januarie, Februarie en Maart
- Tydperk vir kwartaal 2: April, Mei en Junie
- Tydperk vir kwartaal 3: Julie, Augustus en September
- Tydperk vir kwartaal 4: Oktober, November en Desember
- Geen gelentheid is tydens kwartaal 4 gehou nie.
- 'n Waarde van 0 moet teruggestuur word in die gevalle waar 'n ander waarde as 1, 2 of 3 gegenereer is.

(7)



VRAAG 2: JAVA – OBJEK-GEORIENTEERDE PROGRAMMERING

Die graad 11-leerders moet fondse insamel om die graad 12-afskedstfunksie aan die einde van die jaar aan te bied. Programmatuur wat sal help met die bestuur van die inkomste en uitgawes van die verskillende geleenthede wat gehou is om fondse in te samel, is gedeeltelik ontwikkel.

Die gegee program in die **Vraag2_Java**-leërgids bestaan uit 'n klaseenheid, wat 'n fondsinsamelingseleenthede-objek beskryf, en 'n toetsklas, wat 'n skikking van fondsinsamelingseleenthede-objekte skep. Die program gebruik die skikking van objekte om spesifieke inligting wat benodig word, te vertoon.

Die program is ontwerp om die naam van 'n geleentheid, die naam van die persoon wat die geleentheid georganiseer het, die datum waarop die geleentheid plaasgevind het, asook die inkomste ontvang en die uitgawes aangegaan van 'n geleentheid, te stoor.

Doen die volgende:

- Hernoem die leërgids **Vraag2_Java** tot **Vraag2_X** (waar X met jou eksamennummer vervang moet word).

- Hernoem die **Geleentheid.java**-leër in die leërgids **Vraag2_X** tot **GeleentheidXXX.java** (waar XXXX met die laaste VIER syfers van jou eksamennummer vervang moet word).

- Maak die **GeleentheidXXX.java**-leër oop.
- Verander die naam van die klas en die naam van die **konstruktormetode** na **GeleentheidXXX** (waar XXXX met die laaste VIER syfers van jou eksamennummer vervang moet word).

- Voeg jou eksamennummer by as kommentaar in die eerste reël van die **GeleentheidXXX.java**-klas. Stoor die leër.

- Hernoem die **ToetsVraag2.java**-leër in die leërgids **Vraag2_X** tot **ToetsVraag2XXX.java** (waar XXXX met die laaste VIER syfers van jou eksamennummer vervang moet word).

- Maak die **ToetsVraag2XXX.java**-leër oop.
- Verander die naam van die klas na **ToetsVraag2XXX** (waar XXXX deur die laaste VIER syfers van jou eksamennummer vervang moet word). Stoor die leër.

- 2.1 Die objekklas met die naam **GeleentheidXXX.java** sal 'n enkele fondsinsamelingseleenthede beskryf.

Let op die volgende:

- **Gebruik die veldname wat in VRAAG 2.1.1 in hierdie klas aangedui is.** Nie een van die veldde in hierdie klas mag vir die toetsklas (toepassing) toeganklik wees nie.
- Al die metodes van hierdie klas moet vir die toetsklas (toepassing) toeganklik wees.
- Kode **moet nie herhaal word** om 'n gegee probleem op te los nie. Waar moontlik, **moet** die gegee metodes gebruik word om take uit te voer wat nodig is om probleme op te los.
- 'n Deel van die kode in die gegee klas is as kommentaar ingevoeg sodat die klas kan kompilleer. Daar sal van jou verwag word om die kommentaartekens te verwyder, soos aangedui in die vrae wat volg.



Voorbeeld van die afvoer, indien **Rys** die item is wat deur die gebruiker ingesleutel word:

Totaal
=====
R 5135.00

(5)

1.5 Mnr. Du Plessis (die graad 12B's se onderwyser) wil weet wat die totale bedrag is wat ontvang is in terme van donasies aan die graad 12B-klas vir elkeen van die onderskeie items. Voltooi die kode vir **Opsie E** deur 'n SQL-stelling te formuleer wat die totale bedrag wat vir elke item ontvang is vir die stalletjie wat deur die graad 12B's bedryf word, sal vertoon. Gebruik **BedragOntvang** as die nuwe veld se naam.

Voorbeeld van die afvoer:

Item	BedragOntvang
=====	
Ander uitgawes	R 105.00
Groente	R 150.00
Rys	R 195.00
Vleis	R 100.00

(7)

1.6

Mnr. Sweety is ingelig dat die geskatte getal gaste vir al die A-klasse (dit is graad 8A, 9A, 10A, 11A en 12A) met 5% toegeneem het volgens die kaartjieverkope. Voltooi die kode vir **Opsie F** deur 'n SQL-stelling te formuleer wat die **tblDonasies**-tabel hiervolgens sal **opdateer**.

Voorbeeld van die afvoer:

Rekords Suksesvol Verwerk

(5) **WENK:** Voer **Opsie A** uit om te verifieer dat die rekords opgedateer is.

1.7

Een van die ouers van 'n graad 12B-leerder het R200,00 vir 'ander uitgawes' aan die klas geskenk. Graad 12B bedryf die stalletjie met ID-nommer HC77 (**StalletjieID**). Voltooi die kode vir **Opsie G** deur 'n SQL-stelling te formuleer wat die donasie by die rekords sal voeg.

Voorbeeld van die afvoer:

Rekords Suksesvol Verwerk

(4) **WENK:** Voer **Opsie E** uit om te verifieer dat die rekord opgedateer is.

- Tik jou eksamennummer in as kommentaar in die eerste reël van die lêer met die naam **ToetsVraag1XXXX.java** wat die SQL-stellings bevat.
- Stoor die **ToetsVraag1XXXX.java**-lêer.
- 'n Drukstuk van die kode van die **ToetsVraag1XXXX.java**-lêer sal moontlik vereis word (sien instruksie 10 op bladsy 3).

[33]



1.1

Mnr. Sweety wil 'n lys van die geskatte getal gaste wat die stallejies gaan besoek, hê. Voltol die kode vir **Opsie A** deur 'n SQL-stelling te formuleer wat die naam van die stallejies, klas en getal gaste van al die stallejies in die **tblStallejies**-tabel sal vertoon. Vertoon die resultate in dalende volgorde volgens die getal gaste.

Voorbeeld van die afvoer van die eerste vyf rekords:

StallejiesNaam	Klas	GetalGaste
Crazy Dolls	9B	144
Gaming Palace	8A	142
The Rock	11A	136
The W	9C	126
Double Espresso	9E	122

:

(3)

1.2

Die onderhoof, me. Shelby, moet al die onderwysers wat verantwoordelik is vir die stallejies op die C-rugbyveld, met 'n verwagte getal gaste van 100 of meer, kontak. Voltol die kode vir **Opsie B** deur 'n SQL-stelling te formuleer wat die name van al die onderwysers wat vir hierdie stallejies verantwoordelik is, sal vertoon.

Voorbeeld van die afvoer:

Onderwyser
=====
Ferreira, G
Fouche, JC

(4)

1.3

Mnr. Sweety moet die geskatte getal porsies wat vir elke stallejies berei moet word, bereken, deur 'n syfer van 1,25 porsies per verwagte gas te gebruik. Voltol die kode vir **Opsie C** deur 'n SQL-stelling te formuleer wat die naam van die stallejies, die getal gaste wat verwag word en die getal porsies om vir elke stallejies te berei in die **tblStallejies**-tabel sal vertoon. Die getal porsies om te berei is 'n berekende veld en moet tot die naaste heelgetal afgerond word. Gebruik **PorsiesOmTeBerei** as die veld se naam.

Voorbeeld van die afvoer vir die eerste vyf rekords:

StallejiesNaam	GetalGaste	PorsiesOmTeBerei
Gaming Palace	142	178.0
Blockbusters	12	15.0
Green Hills	110	138.0
Twenty Something	88	110.0
Blue Pilots	56	70.0

:

(5)

1.4

Voltol die kode vir **Opsie D** deur 'n SQL-stelling te formuleer wat die totale bedrag donasies wat vir 'n sekere item belowe is, maar wat nog nie ontvang is nie, te vertoon. Laat die gebruiker toe om die naam van die item vanaf die sleutelbord in te sleutel. Gebruik **Totaal** as die berekende veld se naam.



Doen die volgende:

- Hernoem die lêergids **Vraag1_Java** tot **Vraag1_X**, waar X met jou eksamennummer vervang moet word.
- Maak die onvolledige program **ToetsVraag1_Java** in die **Vraag1_X**-lêergids oop.
- Verander die naam van die klas na **ToetsVraag1XXXX** (waar XXXX met die laaste VIER syfers van jou eksamennummer vervang moet word).
- Stoor die klas as **ToetsVraag1XXXX_Java** (waar XXXX met die laaste VIER syfers van jou eksamennummer vervang moet word).

Die konneksiekode, asook die kode om die resultate te vertoon, is reeds as deel van die gegewe kode in die lêer met die naam **Fondse_Java** geskryf.

NOTA: As jou program nie met die databasis kan koppel nie, maak seker dat die databasislêer **FondseDB** in dieselfde lêergids as jou program is. As dit nie die geval is nie, kopieer die databasislêer **FondseDB** na dieselfde lêergids as jou program. Jou program sal nie werk as die databasislêer in 'n ander lêergids as jou program is nie.

NOTA: As jy nog steeds nie konektiwiteit met die databasis kan bewerkstellig wanneer jy die program uitvoer nie, moet jy nogtans die SQL-kode doen en dit inlewer om nagesien te word.

Punte sal slegs toegeken word vir die programmeringskode wat die SQL-stellings in die lêer met die naam ToetsVraag1XXXX_Java bevat.

Wanneer jy die **ToetsVraag1XXXX_Java**-lêer kompilleer en uitvoer, sal die keuselys hieronder vertoon word. As jy egter enigeen van die opsies (A tot G) insleutel, sal die program nie werk nie as gevolg van die onvolledige SQL-stellings.

KEUSELYS	
Opsie A	
Opsie B	
Opsie C	
Opsie D	
Opsie E	
Opsie F	
Opsie G	
V - VERLAAT	
Jou Keuse? [

Doen die volgende:

Voltooi die SQL-stellings in die **ToetsVraag1XXXX_Java**-lêer vir elke opsie op die keuselys, soos in VRAAG 1.1 tot 1.7 hieronder aangedui. Die kode wat die SQL-stellings na die betrokke metodes in die **Fondse_Java**-lêer stuur, is aan jou gegee. Jy moet slegs die SQL-stellings en 'n paar toevoerstellings voltooi, soos in die **ToetsVraag1XXXX_Java**-lêer vereis word.



AFDELING B

Beantwoord AL die vrae in hierdie afdeling slegs as jy **Java** bestudeer het.

SCENARIO

Fondsinsameling het 'n integrale deel van 'n skool se jaarlikse program geword. Die gemeenskap, in samewerking met die onderwysers en leëders van Sonop Hoërskool, het kragtige saamgespan om verskeie geleenthede deur die loop van die jaar te organiseer om te verseker dat die skool voldoende fondse het om die skool se akademiese, sport- en kulturele aktiwiteite te ondersteun.

VRAAG 1: JAVA – PROGRAMMERING EN DATABASIS

Een van die fondsinsamelingsgeleenthede wat die skool besluit het om te hou, is 'n kerrie-en-rys-dag. Mnr. Sweety, die fondsinsamelingskoördineerder by die skool, het 'n databasis geskep met die naam **FondseDB** om die data vir die kerrie-en-rys-dag te stoor.

Elke klas is verantwoordelik vir 'n stalletjie wat of op rugbyveld A, B of C of op hokkieveld A, B of C opgerig sal word. Spasie word toegeken aan elke stalletjie volgens die geskatte getal gaste wat die stalletjie gaan besoek. Donasies word ingesamel vir die aankoop van sekere van die benodigdhede (vleis, rys, groente of ander items) wat nodig is vir die bereiding en bediening van die kerrie en rys.

'n Onvolledige program is ontwikkel om navrae oor die data in die gegewe databasis te verwerk. Jou taak is om hierdie program te voltooi.

Die databasis met die naam **FondseDB**, asook 'n onvolledige Java-program, is in die lêergids met die naam **Vraag1_Java** gestoor. Die lêergids bevat 'n toetsklas met die naam **ToetsVraag1_Java** en 'n objekklas met die naam **Fondse.java** wat die resultate van die navrae sal vertoon.

NOTA: Die ontwerp van die tabelle in die **FondseDB**-databasis en die voorbeelddata vir hierdie vraag kan in **BYLAE A: Bladsy met beskrywings van tabelle** gevind word.

NOTA: As jy nie die databasis wat voorsien is, kan gebruik nie, volg die instruksies in **BYLAE B** om die databasis te skep voordat jy enige van VRAAG 1.1 tot 1.7 beantwoord.

NOTA: Maak 'n kopie van die **FondseDB**-databasis VOORDAT jy met die oplossing begin. Jy sal 'n kopie van die oorspronklike databasis nodig hê om jou program deeglik te toets.



=====	
Getaille nie gekies nie	
=====	
12:40	
20:0	
13:0	
15:0	
9:0	
19:50	
10:0	
8:90	
11:0	
1:0	
5:30	
16:0	
14:100	
17:0	
18:20	
6:0	
7:0	
=====	
Deelnemer se keuses	
=====	
2:20	
3:0	
4:0	
45 ONGELDIGE GETAL	
2 ALREEDS GEKIES	
Punte: 10	
Prys: Teddliebeer	

- Maak seker dat jou eksamennummer as kommentaar in die eerste reël van die hoofeenheid ('main unit') **Vraag3_UXXXX** ingetik is.
- Stoor al die lêers ('File/Save All').
- 'n Drukstuk van die kode van die eenheid ('unit') **Vraag3_UXXXX** sal moontlik vereis word (sien Instruksie 10 op bladsy 3).

TOTAAL AFDELING A: 120

[38]



EASTERN CAPE

=====	
Getalles nie gekies nie	
=====	
12:40	
20:0	
13:0	
3:0	
15:0	
9:0	
10:0	
8:90	
11:0	
1:0	
16:0	
4:0	
17:0	
18:20	
7:0	
=====	
Deelnemer se keuses	
=====	
2:20	
5:30	
19:50	
14:100	
6:0	
Punte: 200	
Prys: Visslok	

Voorbeeld 2:
Toevoer:

Die volgende getalle word een vir een ingesleutel: **2; 3; 4; 45; 2.**
Afvoer (op die volgende bladsy):



3.2

Vertoon die volgende:

- Indien 'n getal wat buite die interval van 1 tot 20 val, ingesleutel word:
 - Vyf punte moet elke keer van die totale punte van die deelnemer afgetrek word.
 - Die inskrywing word as 'getal ONGELDIGE GETAL' in die nuwe skikking gestoor.
- Voorbeeld: Indien 21 ingesleutel word, sal die inskrywing in die nuwe skikking '21 ONGELDIGE GETAL' wees.

(26)

- Die oorblywende inskrywings in die skattejag na die deelnemer se keuses
- Die deelnemer se keuses, die punte behaal en die ooreenstemmende punte

Die punte wat aan die deelnemer toegeken word, sal van die totale getal punte wat behaal is, afhang.

Totale Punte	Prys
<= 0	Geen prys
1-100	Teddiebeer
101-200	Visstok
> 200	Gimnasiumlidmaatskap

(12)

Voorbeeldtoetslopie:

Voorbeeld 1:

Toevoer:

Die volgende getalle word een vir een ingesleutel: 2; 5; 19; 14; 6.

Afvoer (op die volgende bladsy):





'n Lys van getalle van 1 tot 20 en hul ooreenstemmende punte word in 'n skikking met die naam **arrPunte** gestoor.

NOTA: Die 20 elemente in die skikking is nie in numeriese volgorde nie.

Voorbeeld van die verklaring van die skikking **arrPunte**:

```
arrPunte : Array[1..20] of string =
('12:40', '20:0', '13:0', '13:0', '15:0', '9:0', '19:50', '10:0',
'8:90', '11:0', '1:0', '5:30', '16:0', '14:100', '4:0', '17:0',
'18:20', '6:0', '7:0', '2:20');
```

Die formaat van elke inskrywing in die skikking is soos volg:

'12:40': **12** verwys na die getal (wat die deelnemer sal kies) en **40** verwys na die aantal punte wat agter die getal 12 weggesteek is.

'20:0': **20** verwys na die getal (wat die deelnemer sal kies) en **0** verwys na die aantal punte wat agter die getal 20 weggesteek is, ensovoorts.

Voltooi die kode vir die **Speel**-knoppie om die volgende te doen:

3.1 Laat die deelnemer toe om vyf getalle in te sleutel.

Skep 'n skikking om die inskrywings wat deur die gebruiker ingesleutel is, te stoor.

Die volgende moet vir elke getal wat ingesleutel is, gedoen word:

- Tel die ooreenstemmende punte vir die getal wat ingesleutel is, by die totaal van die deelnemer.
- Verwyder die inskrywing uit die gegewe skikking. Voorbeeld: Indien 8 ingesleutel word, moet die inskrywing '8:90' uit die **arrPunte**-skikking ingesleutel word.
- Stoor die inskrywing in 'n **nuwe skikking** in dieselfde formaat as in die gegewe skikking. Voorbeeld: Indien 8 ingesleutel word, moet die inskrywing '8:90' in die nuwe skikking gestoor word.
- Indien 'n getal meer as een keer ingesleutel word:
 ○ Vyf punte moet elke keer van die totale punte van die deelnemer afgetrek word.
- Die inskrywing word as 'getal ALREEDS GEKIES' gestoor.

Voorbeeld: Indien 8 vir 'n tweede keer ingesleutel word, sal die inskrywing in die nuwe skikking '8 ALREEDS GEKIES' wees.



VRAAG 3: DELPHI – PROGRAMMERING

'n Skattejag is een van die aktiwiteite wat as fondsinwingsprojek gebruik word tydens 'n jaarlikse markdaggeleentheid by die skool.

Daar word van deelnemers verwag om 'n inskrywingsfooi te betaal en vyf verskillende getalle van 1 tot 20 te kies. Punte word toegeken aan elke getal wat gekies is. Afhangend van die totale getal punte wat 'n deelnemer het, word verskillende pryse toegeken.

Reëls van die speljie:

- Die getalle 1 tot 20 word gebruik om uit te kies tydens die skattejag vir deelnemers.
- 'n Aantal punte is agter elke getal weggesteek.
- Die deelnemer moet vyf verskillende getalle in die interval van 1 tot 20 insleutel.
- Die weggesteekte aantal punte wat toegeken is aan elke getal wat ingesleutel is, sal opgetel word om die totale aantal punte van die deelnemer te bepaal.
- Die deelnemer word nie toegelaat om dieselfde getal meer as een keer in te sleutel nie. Elke keer wanneer 'n vorige ingesleutelde getal weer gekies word, word vyf punte van die totale getal punte van die deelnemer afgetrek.
- Elke keer wanneer 'n getal buite die interval van 1 tot 20 ingesleutel word, word vyf punte van die totale getal punte van die deelnemer afgetrek.
- 'n Prys word aan die deelnemer toegeken afhangend van die totale getal punte wat behaal is.

LET WEL: Hierdie is 'n probleemoplossingsvraag. Die volgende is van toepassing:

- Jy moet jou eie oplossing volgens die spesifikasies in die vraestel ontwikkel.
- Goëie programmeringsbeginsels moet gevolg word, byvoorbeeld beskrywende veranderlikename, inkeping, ensovoorts.
- 'n Moduliëre programmeringsbenadering moet gevolg word. 'n Maksimum van **vier punte** kan afgetrek word indien jou oplossing nie moduliër is nie, dit wil sê dit het geen subprogramme met paramateroordrag nie.
- Die program moet slegs vir EEN deelnemer te werk.

Jy is voorsien van 'n onvolledige program in die **Vraag3_Delphi**-leërgids ('folder').

Doen die volgende:

- Hernoem die leërgids met die naam **Vraag3_Delphi** tot **Vraag3_X** (waar X met jou eksamenommer vervang moet word).
- Maak Delphi oop en maak dan die lêer met die naam **Vraag3_P.dpr** in die **Vraag3_X**-leërgids oop.
- Gaan na 'File/Save As ...' en stoor die eenheid ('unit') as **Vraag3_UXXXX** (waar XXXX met die laaste VIER syfers van jou eksamenommer vervang moet word).
- Gaan na 'File/Save Project As ...' en stoor die projek as **Vraag3_PXXXX** (waar XXXX met die laaste VIER syfers van jou eksamenommer vervang moet word).
- Voeg jou eksamenommer by aan die regterkant van die opskrif ('caption') van die vorm.
- Voer die program uit. Die volgende koppelvlak sal vertoon word (op die volgende bladsy):



2.2.4

Keuselys Opsie C

Wanneer die gebruiker op hierdie opsie klik, moet die program die **vindkwartaal**- en die **berekenWins**-metodes gebruik om die totale wins wat gedurende elke kwartaal gemaak is, te bepaal en te vertoon.

Voorbeeld van die afvoer:

Wins per kwartaal		
Kwartaal 1	Kwartaal 2	Kwartaal 3
R 6,820.00	R 2,895.00	R 9,725.00

(8)

- Maak seker dat jou eksamennummer as kommentaar in die eerste reël van die hoofeenheid **Vraag2_UXXXX.pas**, asook die klaseenheid **uGeleentheidXXXX.pas**, ingetik is.
- Stoor al die lêers ('File/Save All').
- Drukstukke van die kode van die eenhede **Vraag2_UXXXX.pas** en **uGeleentheidXXXX.pas** sal moontlik vereis word (sien Instruksie 10 op bladsy 3).

[49]



LET WEL: Die program moet die inligting van 'n onbekende aantal geleenthede uit die teksleer kan lees.

Skryf kode in die **OnActivate-gebeurtenishanteerder** ('event handler') van die vorm om die inligting soos volg uit die teksleer **Data2011A.txt** te lees:

Toets of die teksleer bestaan. Vertoon 'n gepaste boodskap indien die leer nie bestaan nie en beëindig die program.
Indien die leer bestaan, gebruik 'n lus om die inligting volgens die volgende stappe uit die teksleer te lees:

- Lees die eerste reël uit die leer en breek die teks op in die naam van die geleentheid, die naam van die organiseerder en die datum van die geleentheid.

- Lees die volgende twee reëls uit die teksleer as die inkomste en die uitgawes van die geleentheid.

- Gebruik hierdie inligting om 'n nuwe **Geleentheid**-objek te skep en plaas die objek in die skikking.

- Gebruik 'n teller om rekord te hou van hoeveel objekte in die skikking geplaas is.

2.2.2 Keuselys Opsie A

Wanneer die gebruiker op hierdie opsie klik, moet die program alle inligting oor al die geleenthede wat in die skikking gestoor is, vertoon.

Voltooi die kode om die afvoer soos volg te vertoon:

Naam	Organiseerder	Datum	Inkomste	Uitgawes
Preloopting#1	Mrs. Jackson, Harold James	2011/05/12	R 2,500.00	R 500.00
Basaar	Mrs. Freeman, Jane	2011/02/24	R 750.00	R 0.00
Uitloopting#2	Mrs. Green, Lee-Ann Suzanne	2011/04/26	R 780.00	R 300.00
Uitloopting#2	Mrs. Ntshumbe, Eric	2011/03/11	R 375.00	R 0.00
LA-N-Spetletjieskonferensie	Mrs. Xaba, Mary Elizabeth	2011/08/01	R 2,400.00	R 600.00
Uitloopting#3	Mrs. Wilson, Jeremy Peter Wayne	2011/07/25	R 425.00	R 0.00
Pannekoekeverkoping	Mrs. Moodley, Harriet Charlene	2011/06/02	R 565.00	R 150.00
Orkese in Aksie	Mrs. Anderson, Brandon Wesley	2011/02/18	R 8,900.00	R 3,000.00
Talentkonferensie	Mrs. Mohammed, Judith Diane Sheryl	2011/07/15	R 9,500.00	R 2,000.00

2.2.3 Keuselys Opsie B

Wanneer die gebruiker op hierdie opsie klik, moet die program 'n opskrif vertoon en die relevante metode roep om die naam van die organiseerder en die wins wat met elke geleentheid gemaak is, te vertoon. Die naam van die organiseerder moet in die volgende formaat vertoon word:

Voorletters<spasie>Van

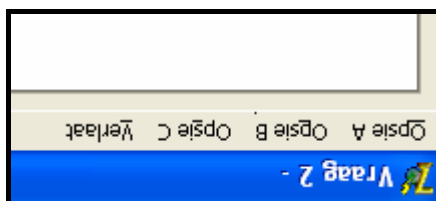
Voorbeeld van die afvoer (op die volgende bladsy):



EASTERN CAPE

2.2

In die **Vraag2_UXXXX**-eenheid (die hooftseenheid) is kode aan jou gegee om die volgende keuselys te vertoon wanneer jy die program uitvoer:



Maak die **Vraag2_UXXXX**-eenheid (die hooftseenheid) oop.

Voeg jou eksamennummer as kommentaar in die eerste reël van die **Vraag2_UXXXX**-eenheid by.

2.2.1 'n Skikking om 'n maksimum van twintig **TGeleentheid**-tipe objekte te stoor, is verklaar.

'n Tekst lêer met die naam **Data2011A.txt**, wat inligting bevat oor al die fondsinasamelingseleenthede wat gedurende die jaar plaasgevind het, is in die **Vraag2_X**-leërgids verskat. Die inligting oor elke geleentheid is oor drie reëls in die volgende formaat in die tekst lêer gestoor:

- **Eerste reël:** Die naam van die geleentheid gevolg deur 'n dubbelpunt (:) en die naam van die organiseerder gevolg deur die datum waarop die geleentheid gehou is. Let daarop dat die naam van die organiseerder en die datum waarop die geleentheid gehou is, deur die woord 'op' geskei word.
- **Tweede reël:** Die inkomste as 'n bedrag
- **Derde reël:** Die uitgawes as 'n bedrag

Voorbeeld van die inhoud van die tekst lêer:

```

Pretloop:Mnr. Jackson, Harold James op 2011/05/12      2500
Uitlotting#1:Mev. Freeman, Jane op 2011/02/24        500
Basaar:Mev. Green, Lee-Ann Suzanne op 2011/04/26     780
Uitlotting#2:Mnr. Nthumba, Eric op 2011/03/11        300
LAN-spelsetjieskompetisie:Mev. Xaba, Mary Elizabeth op 2011/08/01 2400
Uitlotting#3:Mnr. Wilson, Jeremy Peter Wayne op 2011/07/25 425
Pannekoekverkoping:Mev. Moodley, Harriot Charlene op 2011/06/02 565
Orkeste in Akste:Mnr. Anderson, Brandon Westley op 2011/02/18 8900
Talentkompetisie:Mev. Mohammed, Judith Diane Sheryl op 2011/07/15 9500
2000

```



Daar word van jou verwag om die gegewe kode in die **uGeenheidXXXX**-eenheid te korrigeer en te voltooi deur die volgende te doen:

2.1.1 Die konstruktor ('constructor') ontvang die volgende inligting oor 'n geleentheid as parameters:

- **Naam**, wat na die naam van die funksie verwys
- **Persoon**, wat na die naam van die persoon wat die funksie georganiseer het, verwys
- **Datum** waarop die funksie plaasgevind het
- **Inkoste** wat gegenereer is
- **Uitgawes** wat aangegaan is

Die klasveranderlikes is in die konstruktor geïnisialiseer deur die parameterwaardes te gebruik. Die klasveranderlikes is egter nog nie verklaar nie.

Doen die volgende:

- Verklaar die klasveranderlikes van die klas deur die veranderlikename wat in die konstruktor gebruik is, te gebruik.
- Verwyder die kommentaartekens uit die stellings wat die klasveranderlikes in die konstruktor inisialiseer.
- Verwyder die kommentaartekens uit die terugstuurstelling in die gegewe **toString**-metode.

(5)

2.1.2 Skryf 'n metode met die naam **berekenWins** om die wins wat met die geleentheid gemaak is, te bereken en terug te stuur.

(3)

2.1.3 Skryf 'n metode met die naam **vindKwartaal** wat 'n waarde (1, 2, 3 of 0), wat die kwartaal aandui waartydens die geleentheid plaasgevind het, sal bepaal en terugstuur.

Die volgende is van toepassing:

- Tydperk vir kwartaal 1: Januarie, Februarie en Maart
- Tydperk vir kwartaal 2: April, Mei en Junie
- Tydperk vir kwartaal 3: Julie, Augustus en September
- Tydperk vir kwartaal 4: Oktober, November en Desember
- Geen geleentheid is tydens kwartaal 4 gehou nie.
- 'n Waarde van 0 moet teruggestuur word in die gevalle waar 'n ander waarde as 1, 2 of 3 gegenereer is.

(7)

2.1.4 Skryf 'n metode met die naam **skepNaamString** wat 'n nuwe string sal saamstel en terugstuur wat die naam van die organiseerder van 'n geleentheid in die volgende formaat sal bevat:

Voorletters<spasie>Van

Voorbeeld:

Indien die naam van die persoon mnr. Johnson, John Patrick, is, moet die nuwe string **JP Johnson** wees.

(10)



VRAAG 2: DELPHI – OBJEK-GEORIENTEERDE PROGRAMMERING

Die graad 11-leerders moet fondse insamel om die graad 12-afskedstfunksie aan die einde van die jaar aan te bied. Programmatuur wat sal help met die bestuur van die inkomste en uitgawes van die verskillende geleenthede wat gehou is om fondse in te samel, is gedeeltelik ontwikkel.

Die gegewe program in die **Vraag2_Delphi**-leërgids bestaan uit 'n klaseenheid, wat 'n fondsinsamelingseleenthede-objek beskryf, en 'n hoofeenheid, wat 'n skikking van fondsinsamelingseleenthede-objekte skep. Die program gebruik die skikking van objekte om spesifieke inligting wat benodig word, te vertoon.

Die program is ontwerp om die naam van 'n geleentheid, die naam van die persoon wat die geleentheid georganiseer het, die datum waarop die geleentheid plaasgevind het, asook die inkomste ontvang en die uitgawes aangegaan van 'n geleentheid, te stoor.

Doen die volgende:

- Hernoem die leërgids **Vraag2_Delphi** tot **Vraag2_X** (waar X met jou eksamennummer vervang moet word).

- Maak Delphi oop en maak dan die lêer **Vraag2_P.dpr** in die lêergids **Vraag2_X** oop.

- Gaan na 'File/Save As ...' en stoor die eenheid ('unit') as **Vraag2_UXXXX** (waar XXXX met die laaste VIER syfers van jou eksamennummer vervang moet word).

- Maak die eenheid ('unit') **uGeleentheid.pas** oop.

- Gaan na 'File/Save As ...' en stoor die eenheid ('unit') as **uGeleentheidXXXX.pas** (waar XXXX met die laaste VIER syfers van jou eksamennummer vervang moet word).

- Gaan na 'File/Save Project As ...' en stoor die projek as **Vraag2_PXXXX** (waar XXXX met die laaste VIER syfers van jou eksamennummer vervang moet word).

2.1 Die klaseenheid met die naam **uGeleentheidXXXX.pas** bevat 'n

TGeleentheid-klas wat 'n enkele fondsinsamelingseleenthede beskryf.

Let op die volgende:

- **Gebruik die veldname wat in VRAAG 2.1.1 in hierdie klas aangedui is.** Nie een van die veldde in hierdie klas mag vir die hoofeenheid (vormeenheid) toeganklik wees nie.
- Al die metodes van hierdie klas moet vir die hoofeenheid (vormeenheid) toeganklik wees.

- Kode **moet nie herhaal word** om 'n gegewe probleem op te los nie. Waar moontlik, **moet** die gegewe metodes gebruik word om take uit te voer wat nodig is om probleme op te los.

- 'n Deel van die kode in die gegewe klas is as kommentaar ingevoeg sodat die eenheid kan kompilleer. Daar sal van jou verwag word om die kommentaartekens te verwyder, soos aangedui in die vrae wat volg.

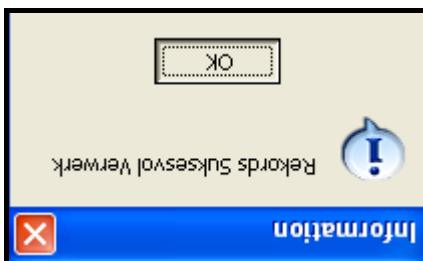
- Bykomend tot die wysiging van die gegewe metodes sal daar van jou verwag word om kode vir nuwe metodes by te voeg, soos beskryf in die vrae wat volg.



1.7

Een van die ouers van 'n graad 12B-leerder het R200,00 vir 'ander uitgawes' aan die klas geskenk. Graad 12B bedryf die stalletjie met ID-nommer HC77 (**StalletjieID**). Voltooi die kode vir die **Opsie G**-knoppie ('button') deur 'n SQL-stelling te formuleer wat die donasie by die rekords sal voeg.

Voorbeeld van die afvoer:



WENK: Kies **Opsie E** om te verifieer dat die rekords opdateer is.

(4)

- Tik jou eksamennummer in as kommentaar in die eerste reël van die lêer met die naam **Vraag1_UXXXX.pas** wat die SQL-stellings bevat.
- Stoor die eenheid ('unit') **Vraag1_UXXXX** en die projek **Vraag1_PXXXX** ('File/Save All').
- 'n Drukstuk van die kode van die **Vraag1_UXXXX.pas**-lêer sal moonklik vereis word (sien Instruksie 10 op bladsy 3).

[33]



1.4

Voltooi die kode vir die **Opsie D**-knoppie ('button') deur 'n SQL-stelling te formuleer wat die totale bedrag donasies wat vir 'n sekere item belowe is, maar wat nog nie ontvang is nie, te vertoon. Laat die gebruiker toe om die naam van die item vanaf die sleutelbord in te sleutel. Gebruik **Totaal** as die berekende veld se naam.

Voorbeeld van die afvoer, indien **Rys** die item is wat deur die gebruiker ingesleutel word:

Totaal	R 5,135.00
---------------	------------

(5)

1.5

Mnr. Du Plessis (die graad 12B's se onderwyser) wil weet wat die totale bedrag is wat ontvang is in terme van donasies aan die graad 12B-klas vir elkeen van die onderskeie items. Voltooi die kode vir die **Opsie E**-knoppie ('button') deur 'n SQL-stelling te formuleer wat die totale bedrag wat vir elke item ontvang is vir die stalleijie wat deur die graad 12B's bedryf word, sal vertoon. Gebruik **BedragOntvang** as die nuwe veld se naam.

Voorbeeld van die afvoer:

Item	BedragOntvang
Ander uitgawes	R 105.00
Groente	R 150.00
Rys	R 195.00
Vleis	R 100.00

(7)

1.6

Mnr. Sweety is ingelig dat die geskatte getal gaste van al die A-klasse (dit is graad 8A, 9A, 10A, 11A en 12A) met 5% toegeneem het volgens die kaartjieverkope. Voltooi die kode vir die **Opsie F**-knoppie ('button') deur 'n SQL-stelling te formuleer wat die **tblDonasies**-tabel hiervolgens sal opdateer.

Voorbeeld van die afvoer:



(5)

WENK: Kies **Opsie A** om te verifieer dat die rekords opgedateer is.



1.1

Mr. Sweety wil 'n lys van die geskatte getal gaste wat die stalletjies gaan besoek, hê. Voltooi die kode vir die **Opsie A**-knoppie ('button') deur 'n SQL-stelling te formuleer wat die **naam van die stalletjie, klas en getal gaste** van al die stalletjies in die **tblStalletjies**-tabel sal vertoon. Vertoon die resultate in dalende volgorde volgens die getal gaste.

Voorbeeld van die afvoer van die eerste vyf rekords:

StalletjieNaam	Klas	GetalGaste
Crazy Dolls	9B	144
Gaming Palace	8A	142
The Rock	11A	136
The Ww	9C	126
Double Espresso	9E	122

:

(3)

1.2

Die onderhoof, me. Shelby, moet al die onderwysers wat verantwoordelik is vir die stalletjies op die C-rugbyveld, met 'n verwagte getal gaste van 100 of meer, kontak. Voltooi die kode vir die **Opsie B**-knoppie ('button') deur 'n SQL-stelling te formuleer wat die **name van al die onderwysers** wat vir hierdie stalletjies verantwoordelik is, sal vertoon.

Voorbeeld van die afvoer:

Onderwyser
Ferreira, G
Fouché, JC

(4)

1.3

Mr. Sweety moet die geskatte getal porsies wat vir elke stalletjie berei moet word, bereken, deur 'n syfer van 1,25 porsies per verwagte gas te gebruik. Voltooi die kode vir die **Opsie C**-knoppie ('button') deur 'n SQL-stelling te formuleer wat die **naam van die stalletjie, die getal gaste** wat verwag word en die **getal porsies om vir elke stalletjie te berei** in die **tblStalletjies**-tabel sal vertoon. Die **getal porsies om te berei** is 'n berekende veld en moet tot die naaste heelgetal afgerond word. Gebruik **PorsiesOmTeBerei** as die veld se naam.

Voorbeeld van die afvoer vir die eerste vyf rekords:

StalletjieNaam	GetalGaste	PorsiesOmTeBerei
Gaming Palace	142	178
Blockbusters	12	15
Green Hills	110	138
Twenty Something	88	110
Blue Pilots	56	70

:

(5)



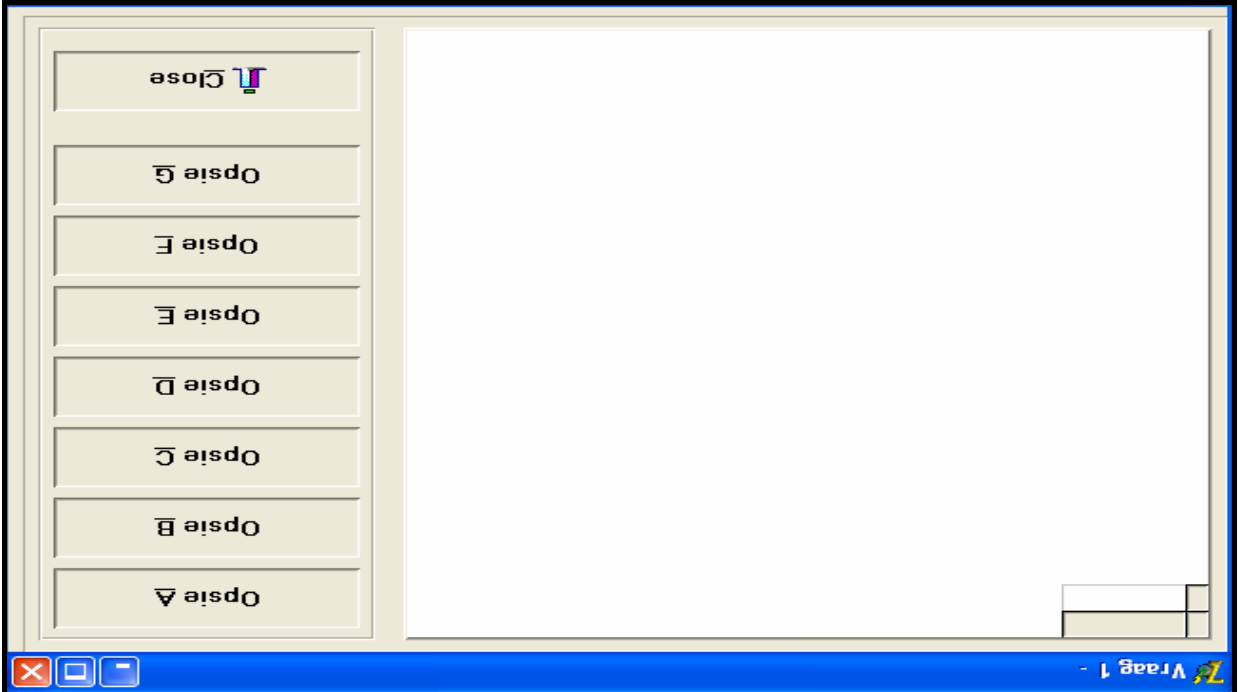
- Gaan na 'File/Save As ...' en stoor die eenheid ('unit') as **Vraag1_UXXXX** (waar XXXX met die laaste VIER syfers van jou eksamenommer vervang moet word).
- Gaan na 'File/Save Project As ...' en stoor die projek as **Vraag1_PXXXX** (waar XXXX met die laaste VIER syfers van jou eksamenommer vervang moet word).
- Die program behoort met die databasis met die naam **FondseDB** te kan koppel. Gebruik die stappe wat in **BYLAE C** voorsien is om konektiwiteit met die databasis te bewerkstellig wanneer jy VRAAG 1.1 (op die volgende bladsy) doen en agterkom dat die konektiwiteit nie werk nie.

NOTA: As jou program nie met die databasis kan koppel nie, maak seker dat die databasisleër **FondseDB** in dieselfde lêerids as jou program is. Jou program sal nie werk as die databasisleër in 'n ander lêerids as die program is nie. As dit die geval is, kopieer die databasisleër **FondseDB** na dieselfde lêerids as jou program.

NOTA: As jy nog steeds nie konektiwiteit met die databasis kan bewerkstellig wanneer jy die program uitvoer nie, moet jy nogtans die SQL-kode doen en dit inlewer om nagesien te word.

Punte sal slegs toegeken word vir die programmeringskode wat die SQL-stellings in die Vraag1_UXXXX-eenheid ('unit') bevat.

Wanneer jy die program uitvoer, sal die koppelvlak hieronder vertoon word. Wanneer jy die knoppies ('buttons') klik, sal 'n fout vertoon word as gevolg van die onvolledige SQL-stellings.



Doen die volgende:

Voltooi die SQL-stellings in **Vraag1_UXXXX.pas** vir elke knoppie ('button'), soos in VRAAG 1.1 tot 1.7 hieronder aangedui. Die kode om die SQL-stellings uit te voer en die resultate in die DBGrid te vertoon, is aan jou gegee. Jy moet slegs die SQL-stellings en 'n paar toevoerstellings voltooi, soos in die **Vraag1_UXXXX**-eenheid vereis word.



AFDELING A

Beantwoord AL die vrae in hierdie afdeling slegs as jy **Delphi** bestudeer het.

SCENARIO

Fondsinsameling het 'n integrale deel van 'n skool se jaarlikse program geword. Die gemeenskap, in samewerking met die onderwysers en leëders van Sonop Hoërskool, het kragtige saamgespan om verskeie geleenthede deur die loop van die jaar te organiseer om te verseker dat die skool voldoende fondse het om die skool se akademiese, sport- en kulturele aktiwiteite te ondersteun.

VRAAG 1: DELPHI – PROGRAMMERING EN DATABASIS

Een van die fondsinsamelingsgeleenthede wat die skool besluit het om te hou, is 'n kerrie-en-rys-dag. Mnr. Sweety, die fondsinsamelingskoördineerder by die skool, het 'n databasis geskep met die naam **FondsEB** om die data vir die kerrie-en-rys-dag te stoor.

Elke klas is verantwoordelik vir 'n stalletjie wat of op rugbyveld A, B of C of op hokkieveld A, B of C opgerig sal word. Spasie word toegeken aan elke stalletjie volgens die geskatte getal gaste wat die stalletjie gaan besoek. Donasies word ingesamel vir die aankoop van sekere van die benodigdhede (vleis, rys, groente of ander items) wat nodig is vir die bereiding en bediening van die kerrie en rys.

'n Onvolledige program is ontwikkel om navrae oor die data in die gegewe databasis te verwerk. Jou taak is om hierdie program te voltooi.

Die databasis met die naam **FondsEB**, asook 'n onvolledige Delphi-projek met die naam **Vraag1_P.dpr**, is in die lêergids ('folder') met die naam **Vraag1_Delphi** gestoor.

NOTA: Die ontwerp van die tabelle in die **FondsEB**-databasis en die voorbeelddata vir hierdie vraag kan in **BYLAE A: Bladsy met beskrywings van tabelle** gevind word.

NOTA: As jy nie die databasis wat voorsien is, kan gebruik nie, volg die instruksies in **BYLAE B** om die databasis te skep voordat jy enige van VRAAG 1.1 tot 1.7 beantwoord.

NOTA: Maak 'n kopie van die **FondsEB**-databasis VOORDAT jy met die oplossing begin. Jy sal 'n kopie van die oorspronklike databasis nodig hê om jou program deeglik te toets.

Doen die volgende:

- Hernoem die lêergids **Vraag1_Delphi** tot **Vraag1_X**, waar X met jou eksamenommer vervang moet word.
- Maak Delphi oop en maak dan die lêer **Vraag1_P.dpr** in die **Vraag1_X**-lêergids oop. Die program vertoon agt knoppies ('buttons'), asook 'n DBGrid wat as 'n afvoercomponent gebruik sal word (sien voorbeeld op die volgende bladsy).
- Voeg jou eksamenommer aan die regterkant van 'Vraag 1 –' in die opskrif ('caption') van die vorm by.



6. Tik jou eksamennummer in as 'n kommentaar in die eerste reël van elke program.
7. Lees AL die vrae aandagtig deur. Moenie meer doen as wat die vrae vereis nie.
8. Gedurende die eksamen mag jy die handleidings wat oorspronklik saam met die apparaat en programmatuur verskat is, gebruik. 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.**
9. Aan die einde van hierdie eksamen sessie moet jy die skryf of CD met al jou werk daarop gestoor, inlewer OF jy moet seker maak dat al jou werk op die hardeskyf/netwerk gestoor is, soos deur die toetsighouer/onderwyser aan jou verduidelik is. Maak seker dat al die lêers gelees kan word.
10. Die toetsighouer sal jou inlig of jy drukstukke van die programmeringskode van al die vrae wat jy gedoen het, moet inlewer.
11. Indien drukwerk vereis word, moet al die drukwerk van programmeringsvrae wat jy gedoen het, binne een uur na die afhandeling van hierdie vraestel plaasvind.
12. Voltooi die inligtingsblad wat aan hierdie vraestel geheg is en lewer dit aan die einde van hierdie eksamen sessie in.



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 eksamen sessie te verlaat nie.
2. Beantwoord AFDELING A (vir Delphi-programmeërs) OF AFDELING B (vir Java-programmeërs).
3. Jy het die lys lêers hieronder nodig om die vrae te beantwoord. Dit sal op 'n stiftie OF 'n CD aan jou verskat word OF die toesighouer/onderwyser sal vir jou sê waar om dit op die hardeskyf van die werkstasie wat jy gebruik, te vind OF in watter netwerklêergids ('network folder') dit is.

VRAAG 1
Delphi:
 FondsDB.mdb
 FondsDB.mdb
 tbiDonasies.txt
 tbiDonasies.txt
 tbiStalleijes.txt
 tbiStalleijes.txt
 ToetsVraag1.java
 Vraag1_P.dpr
 Vraag1_P.res
 Vraag1_U.dfm
 Vraag1_U.pas

Java:
 Fonds.java
 FondsDB.mdb
 tbiDonasies.txt
 tbiStalleijes.txt
 ToetsVraag1.java

VRAAG 2
Delphi:
 Data2011A.txt
 uGeleentheid.pas
 Vraag2_P.dpr
 Vraag2_P.res
 Vraag2_U.dfm
 Vraag2_U.pas

Java:
 Data2011A.txt
 Geleentheid.java
 ToetsVraag2.java

VRAAG 3
Delphi:
 Vraag3_P.dpr
 Vraag3_P.res
 Vraag3_U.dfm
 Vraag3_U.pas

Java:
 ToetsVraag3.java

4. Stoor jou werk met gereelde tussenposes as 'n voorsorgmaatreeël teen kragonderbrekings.
 5. Stoor AL jou oplossings in lêrigdse ('folders') met die vraagnommer en jou eksamennummer as die naam van die lêrigds, byvoorbeeld Vraag2_3020160012.
- Indien jy die lêers hierbo op 'n skyf (CD of stiftie) ontvang het, skryf jou eksamennummer op die etiket.





Hierdie vraestel bestaan uit 31 bladsye, 3 bylaes en 'n inligtingsblad.



TYD: 3 uur

PUNTE: 120

INLIGTINGSTEKNOLOGIE V1
FEBRUARIE/MART 2012

GRAAD 12

**NASIONALE
SENIOR SERTIFIKAT**



basic education
Department:
Basic Education
REPUBLIC OF SOUTH AFRICA