education

Department:
Education
REPUBLIC OF SOUTH AFRICA

NATIONAL SENIOR CERTIFICATE

GRADE 12

INFT.1

INFORMATION TECHNOLOGY P1

NOVEMBER 2009

MARKS: 120

TIME: 3 hours

This question paper consists of 23 pages, 3 annexures and an information sheet.

MORNING SESSION

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 EITHER 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 a network folder. If the files are on a CD issued to you, you need to copy them onto your hard drive.

   QUESTION 1

Delphi:	**Java:**
BandB.mdb	BandB.mdb
Question1_U.pas	testBandB.java
Question1_P.dpr	BandB.java
Question1_U.dfm	tblClients.txt
tblClients.txt	tblOrders.txt
tblOrders.txt	BandB.odb (for Linux users)

   QUESTION 2

Delphi:	**Java:**
Question2_P.dpr	testExtraItem.java
Question2_U.dfm	Extras.txt
Question2_U.pas	
Extras.txt	

   QUESTION 3

Delphi:	**Java:**
Question3_P.dpr	testNumbers.java
Question3_U.dfm	
Question3_U.pas	

   If a disk containing the above files was issued to you, write your examination number and centre 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 number of the question and your examination number as the name of the folder, for example Quest2_3020160012.

6. Type your examination number as a comment in the first line of each program.

7. Read ALL the questions carefully. Do only what is required by the question.

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

9. At the end of this examination session you will be required to hand in the stiffy or CD given to you by the invigilator with your work saved on it or you must make sure that all your work has been saved on the network as explained to you by the invigilator/teacher. Ensure that all files can be read before leaving the computer.

10. You also have to hand in printouts of the programming code for all the questions that you did.

11. All printing of programming questions will take place within an hour of the completion of the examination.

12. Complete the separate information sheet that has been provided with the question paper and hand it to the invigilator at the end of the examination.

SECTION A

Answer this section only if you studied **Delphi.**

Answer ALL the questions in this section.

SCENARIO

The Petersen Group CC has set up Bed and Breakfast guesthouses (B&Bs) in all the towns where Soccer World Cup matches will be played in 2010. They have approached you to do some work to help them manage various parts of their business.

QUESTION 1: DELPHI – PROGRAMMING AND DATABASE

The Petersen Group CC wants to use software to assist their staff in answering queries from management. In the development of the software, a database called **BandB.mdb** has been created. The program is incomplete. Your task will be to complete the program that will be used to answer queries from management.

NOTE: The design and sample data for the database can be found in ANNEXURE A: **Table Description Sheet**.

NOTE: If you cannot use the database provided, use the instructions in ANNEXURE B to create the database before you answer the questions.

You have been given an incomplete Delphi program with a unit named **Question1_U** and a project named **Question1_P** in the folder named **Question 1 Delphi**.

Do the following:

- Rename the folder **Question 1 Delphi** as **Quest1_X**, where X should be replaced with your examination number.
- Open Delphi and then open the file **Question1_P.dpr** in the folder **Quest1_X**. The program displays seven buttons as well as a DBGrid that will be used as an output component (see example on next page).
- Add your examination number to the caption of the form to the right of 'Question 1 –'.
- Go to 'File/Save As …' and save the unit as **Question1_Uxxxx** (where xxxx represents the last FOUR digits of your examination number).
- Go to 'File/Save Project As …' and save the project as **Question1_Pxxxx** (where xxxx represents the last FOUR digits of your examination number).

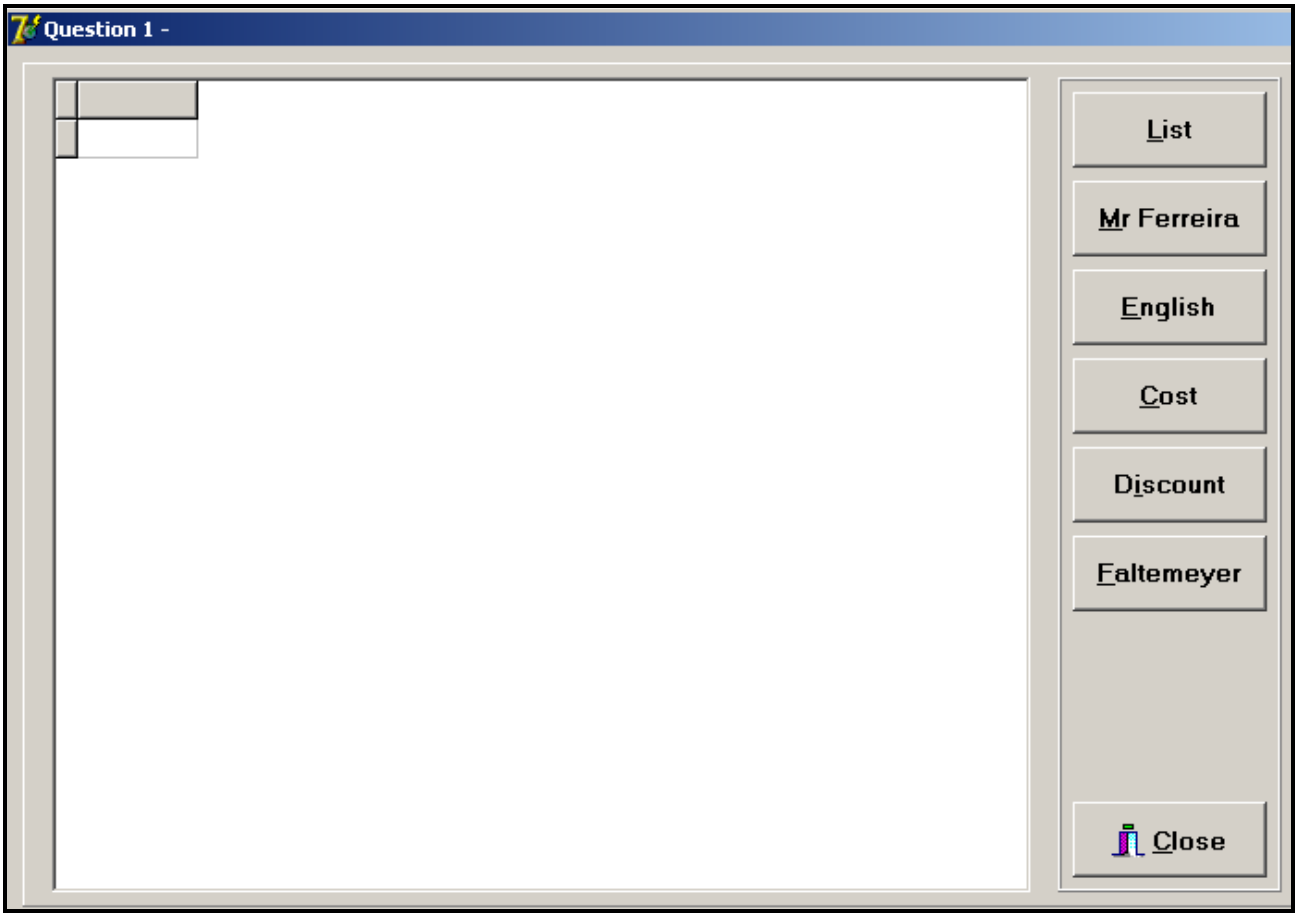
- The program should be able to connect to the database named **BandB.mdb**. When you do QUESTION 1.1 (which follows on the next page) and you find that the connectivity is not in place, use the steps in ANNEXURE C to establish connection with the database.

HINT: If your program cannot connect to the database, make sure that the database file **BandB.mdb** is in the same folder as your program. If not, copy the database file **BandB.mdb** 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 programming code and submit it for marking.

Marks will only be awarded for the program methods that contain the SQL statements in the unit named Question1_Uxxxx.

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



Complete the SQL statements for each button as indicated by QUESTIONS 1.1 to 1.6 that follow.

1.1 Complete the code in the **List** button by formulating an SQL statement to display all the fields from the **tblClients** table sorted by **Surname** and then by **FName** (first name).

Example of output:

ClientNo	Title	Surname	FName	IDNumber	SA	Nationality
10	Ms	Baker	Jean	5612130067143	False	English
21	Mr	Bekker	Boris	7710145678083	False	German
19	Mr	Bekker	Hendrik	5607187367892	True	SA
1	Mr	Ferreira	Guiseppe	3712183567065	False	Portugese
9	Mr	Fouche	Jean-Claude	7805263478653	False	French
5	Mrs	Freulich	Ina	8107166893475	False	German
11	Mr	Griffiths	Daffyd	2311078237876	False	Welsh
2	Mrs	Honeywell	Lucy	5907234676012	True	SA
17	Mr	Johnstone	Paul	8002027876086	False	English
4	Mr	Khoza	Bantu	7706161545342	True	SA
6	Mr	Kowalski	Joachim	7407189162348	False	Polish
20	Ms	Kowalski	Nikitha	8012230067083	False	Polish
7	Ms	Krzyaikis	Stavro	7305172654765	False	Greek
3	Ms	Mendes	Isaac	6708139075176	True	SA
13	Ms	Nguni	Priscilla	7609248768934	True	SA
18	Ms	Rodriguez	Laura	8211263788743	False	Spanish
15	Mr	Saint-Ypres	Yves	6610278547278	False	French
16	Mr	Smythe	Andrew	7410287656786	False	English
12	Mrs	Van Rheede	Roxanne	6312054763457	False	Dutch
8	Ms	Von Braun	Gertrude	5608237896212	False	German
14	Mr	Yamimoto	Ishiguru	7907186427868	False	Japanese

(5)

1.2 Complete the code in the **Mr Ferreira** button by creating a query that will calculate the total amount owed by Mr Ferreira (**ClientNo** field entry is 1). This is a calculated answer. The heading must be **Total Due** and the result must be formatted to display with TWO decimal places.

Example of output:

Total Due
R47.65

(7)

1.3 All the bookings for the English football fans have been cancelled. Complete the code for the **English** button by creating a query to delete all English clients (**Nationality** field entry is English) from the **tblClients** table. (4)

1.4 The group uses a 25% markup when calculating the selling price. Complete the code for the **Cost** button by creating a query that will list the **Date**, **Category, SellingPrice** and **Cost** (selling price minus the 25% markup) for each item ordered by Mr Guiseppe Ferreira (**ClientNo** field entry is 1) from the **tblOrders** table. **Cost** is a calculated field and must be named **Cost**.

Use the following formula to calculate the cost:

Cost = selling price – ((selling price/125)*25)

Example of output:

Date	Category	SellingPrice	Cost
2009/01/25	Phone Call	8.75	7
2009/01/25	Kitchen	29.95	23.96
2009/01/25	Bar	8.95	7.16

(10)

1.5 The Petersen Group has decided that they want to support all soccer fans by giving them R5,00 discount on the selling price of every item they have ordered if the item's selling price is R30,00 or more. Complete the code for the **Discount** button by writing a query that will reduce the selling price of the relevant items by R5,00 in the **tblOrders** table.

NOTE: Every time that you choose the Discount option from the menu the selling prices of items that are still more than R30,00 will be reduced by another R5,00.

Example of output after the Discount option has been executed once:

OrderNo	ClientNo	Date	Category	SellingPrice
1	1	2009/01/25	Phone Call	8.75
2	2	2009/01/25	Bar	84.95
3	3	2009/01/25	Kitchen	44.95
4	1	2009/01/25	Kitchen	29.95
5	1	2009/01/25	Bar	8.95
6	4	2009/01/26	Taxi	154
7	5	2009/01/26	Security	25
8	5	2009/01/26	Internet	14.95
9	6	2009/01/26	Phone Call	22.86

(5)

1.6 Complete the code for the **Faltemeyer** button by writing a query that will add the following client data to the **tblClients** table:

Mr Harald Faltemeyer, ID 7407185683074, Swedish

NOTE: The content of the **IDNumber** field is string and the content of the **SA** field is Boolean. (9)

- Enter your examination number as a comment in the first line of the **Question1_Uxxxx** unit containing the SQL statements.
- Go to 'File/Save All' to save all your work.
- A printout of the code for the **Question1_Uxxxx** unit will be required. **[40]**

QUESTION 2: DELPHI – OBJECT-ORIENTED PROGRAMMING

Guests for the Soccer World Cup want to travel around South Africa to follow their teams. Normally guests have to pay the bill for all the extra items they ordered during their stay when they check out at a guest house. To provide a service which makes them different from (and better than) other B&Bs, the Petersen Group has decided to let their guests transfer their accumulated extra costs between the guest houses in each town. Guests will have to pay the bill for these items when they check out at the last guest house on their journey. They decided that the best way to manage this is to e-mail a text file indicating the extra costs of the guests to the next guest house.

You have been asked to write the program to handle the extra costs of the guests.

The data is stored in a text file named **Extras.txt** in the following format:

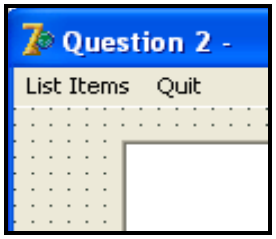
GuestNo#GuestName#ExtraType#CostPerItem

An example of some of the data in the text file:

1#Mr G Ferreira#Phone#7.05
2#Mrs L Honeywell#Drinks#71.95
3#Ms I Mendes#Kitchen#39.95
1#Mr G Ferreira#Kitchen#23.95
1#Mr G Ferreira#Drinks#7.15
4#Mr B Khoza#Taxi#127.25
:

Do the following:

- Rename the folder **Question 2 Delphi** as **Quest2_X** (where X represents your examination number).
- Open Delphi and then open the file **Question2_P.dpr** in the folder **Quest2_X**.
- Go to 'File/Save As …' and save the unit as **testExtraItem_Uxxxx** (where xxxx represents the last FOUR digits of your examination number).
- Go to 'File/Save Project As …' and save the project as **testExtraItem_Pxxxx** (where xxxx represents the last FOUR digits of your examination number).
- The following menu will be displayed when you execute the program:



- Add your examination number to the caption of the form to the right of 'Question 2 –'.

2.1 Create an object class (another unit) named **ExtraItem_Uxxxx** and save this unit as **ExtraItem_Uxxxx** in your **Quest2_X** folder (xxxx should be replaced by the last FOUR digits of your examination number). All fields in this class are private and all methods public. The fields and methods that you need to create and code are described below:

2.1.1 Define a class named **TExtraItem**. Create appropriately named and typed private fields to hold the following data (suggested field names are given in brackets):

- Guest Number (**guestNum**)
- Item type description (**itemType**)
- Cost per item (**cost**) (10 ÷ 2) (5)

2.1.2 Write a constructor method which accepts the guest number, the item description and the cost per item as parameters. All the fields must be initialised in the constructor. (8 ÷ 2) (4)

2.1.3 Write an appropriately named 'get' method (accessor method) to return the guest number. (4 ÷ 2) (2)

2.1.4 The company uses a 25% markup on cost per item to determine profit. Write a method named **calculateProfit** that calculates and returns the profit (that is, cost*25/100). (4 ÷ 2) (2)

2.1.5 Write a method named **calculatePrice** that calculates the final price of the item (that is, cost + the calculated profit). (4 ÷ 2) (2)

2.1.6 Write a method named **toString** that builds and returns a string with information on the item formatted as follows:

Item type<tab>Cost<tab>Profit<tab>Final Price

Values must be formatted to two decimal places. (8 ÷ 2) (4)

2.2 Write code to do the following in the **testExtraItem_Uxxxx** file (the main unit):

2.2.1 Create an array named **arrItems** that holds **TExtraItem** objects. Write code in the **OnActivate Eventhandler** of the form to read information from the text file **Extras.txt** according to the following steps:

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

(b) Use a loop to ...

- read a line of text from the text file.
- separate the text into the guest number, item type and cost.
- use this information to create a new **TExtraItem** object and place the object in the array named **arrItems**.

(c) Use a counter field to keep track of how many items there are in the array. (28 ÷ 2) (14)

2.2.2 Menu Option: List items

When the user selects this menu option the program must do the following:

- Allow the user to enter a guest number
- Search through the array and each time an item for the guest is found …
  - calculate the profit using the percentage mark-up and calculate the final price.
  - display the information using the toString method.
  - add the final price for each item to get a grand total.
- When the search is complete the program must …
  - display the total amount due for the guest.
  - display an appropriate message to say that there are no extra charges for this guest, if no items have been found.

Example of the output:

Information on extra items for guest number 1

Item	Cost	Profit	Price
Phone	R7.05	R1.76	R8.81
Kitchen	R23.95	R5.99	R29.94
Drinks	R7.15	R1.79	R8.94

The total amount due is R47.69

(20 ÷ 2) (10)

- Enter your examination number as a comment in the first line of the main unit **testExtraItem_Uxxxx** as well as the object unit **ExtraItem_Uxxxx**.
- Save all the files ('File/Save All').
- Printouts of the code for the two units (**testExtraItem_Uxxxx** and **ExtraItem_Uxxxx**) will be required.

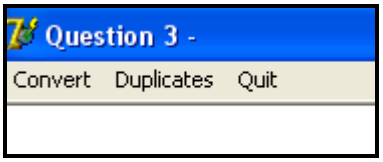
[43]

QUESTION 3: DELPHI PROGRAMMING

You have been given an incomplete program in the folder named **Question 3 Delphi.**

Do the following:

- Rename the folder named **Question 3 Delphi** to **Quest3_X**. Replace X with your examination number.
- Open the Delphi program in this folder.
- Save the unit ('File/Save As') as **Numbers_Uxxxx** and the project ('File/Save Project As') as **Numbers_Pxxxx** inside the folder (xxxx should be replaced by the last FOUR digits of your examination number).
- Add your examination number to the caption of the form to the right of 'Question 3 –'.
- A menu with the following options will be displayed when the program is executed:



The given program generates an array of phone numbers for you to work with. The array called **arrPhoneNos** holds 20 strings. Data in the array will look like this:

086NewHill
086DialBar
086BayView
086KyaSand
086SowetoN
086CasaSol
086TheHavn
086GetFood
086ThaiPlc
086Cleaner
086CasaRok
086RixTaxi
086AirTime
086DialBed
086DialCar
086DialHlp
086KyaRosa
086BaySand
086Cater4U
0861to1Air

3.1 Write code to convert all the alphanumeric characters in the **arrPhoneNos** array into normal telephone numbers and replace them into the same array. Replace the alphabetic characters (upper case and lower case) in the telephone numbers with the corresponding numbers given below:

A, B, C	2
D, E, F	3
G, H, I	4
J, K, L	5
M, N, O	6
P, Q, R, S	7
T, U, V	8
W, X, Y, Z	9

The numeric values in the telephone numbers remain as they are.

NOTE: The resulting numerical phone number must be formatted as follows: 3 digits, space, 3 digits, space, 4 digits (for example 086 345 6546)

The program must display the original alphanumeric number and the new numerical version.

NOTE: Use modular programming techniques when solving this problem.

Example of the output:

Original Number	Converted Number
086NewHill	086 639 4455
086DialBar	086 342 5227
086BayView	086 229 8439
086KyaSand	086 592 7263
086SowetoN	086 769 3866
086CasaSol	086 227 2765
086TheHavn	086 843 4286
086GetFood	086 438 3663
086ThaiPlc	086 842 4752
086Cleaner	086 253 2637
086CasaRok	086 227 2765
086RixTaxi	086 749 8294
086AirTime	086 247 8463
086DialBed	086 342 5233
086DialCar	086 342 5227
086DialHlp	086 342 5457
086KyaRosa	086 592 7672
086BaySand	086 229 7263
086Cater4U	086 228 3748
0861to1Air	086 186 1247

(22)

3.2 Use the numerical phone numbers to check that there are no duplicates in the array. If duplicates are found, the program must display the duplicate numbers. If no duplicates are found, a suitable message must be displayed. At the end of the list there must be a summary stating how many duplicates were found (if any).

Example of the output:

```
Duplicates
086 342 5227
086 227 2765
The number of duplicates : 2
```

Optional: If you wish to further test your program, do the following:

To test for no duplicates, change the given elements of the array as follows:

- Change 086DialBar (the second element in the array) to 086DialBax
- Change 086CasaSol (the sixth element in the array) to 086CasaSox

To test for more than two duplicates, change the given elements of the array as follows:

- Change 086DialBax (the second element in the array) back to 086DialBar
- Change 086CasaSox (the sixth element in the array) back to 086CasaSol
- Replace the last THREE elements in the array with the following numbers:

  086AirVine
  086BayThe9
  086SoxDunn (15)

- Enter your examination number as a comment in the first line of the unit **Numbers_Uxxxx**.
- Save the unit and the project ('File/Save All').
- A printout of the code for the unit **Numbers_Uxxxx** will be required. **[37]**

TOTAL SECTION A: 120

SECTION B

Answer this section only if you studied **Java**.

Answer ALL the questions in this section.

SCENARIO

The Petersen Group CC has set up Bed and Breakfast guesthouses (B&Bs) in all the towns where Soccer World Cup matches will be played in 2010. They have approached you to do some work to help them manage various parts of their business.

QUESTION 1: JAVA – PROGRAMMING AND DATABASE

The Petersen Group CC wants to use software to assist their staff in answering queries from management. In the development of the software, a database called **BandB.mdb** has been created. The program is incomplete. Your task will be to complete the program that will be used to answer queries from management.

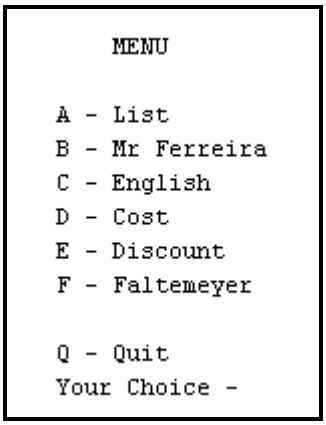
NOTE: The design and sample data for the database can be found in ANNEXURE A: **Table Description Sheet**.

NOTE: If you cannot use the database provided, use the instructions in ANNEXURE B to create the database before you answer the questions.

You have been given a folder named **Question 1 Java** with an incomplete Java program with a test class named **testBandB.java** and an object class named **BandB.java** which will display the results of the queries.

Do the following:

- Rename the folder **Question 1 Java** as **Quest1_X**, where X should be replaced with your examination number.
- Open this folder and rename the **testBandB.java** file as **testBandBxxxx** (where xxxx must be replaced by the last FOUR digits of your examination number).
- Open the incomplete program **testBandBxxxx.java**. Change the name of the class to **testBandBxxxx** (where xxxx must be replaced by the last FOUR digits of your examination number).
- The program will not run because of the incomplete SQL-statements. Once you have entered the correct SQL-statements, the program will display a simple menu with seven options (see next page).



The code for connecting to the database, as well as the code to display the results of the queries, has already been written as part of the given code in the file named **BandB.java**. There is no need for any additional code to establish database connectivity.

HINT: If your program cannot connect to the database, make sure that the database file **BandB.mdb** is in the same folder as your program. If not, copy the database file **BandB.mdb** 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 programming code and submit it for marking.

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

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

Complete the SQL statements in **testBandBxxxx.java** for each menu option as indicated by QUESTIONS 1.1 to 1.6 below. The code to pass the SQL statements to the relevant methods in **BandB.java** has been given. You need only complete the SQL statements.

1.1 Complete the code for menu option A (**List**) by formulating an SQL statement to display all the fields from the **tblClients** table sorted by **Surname** and then by **FName** (first name).

Example of output (on the next page):

ClientNo	Title	Surname	FName	IDNumber	SA	Nationality
10	Ms	Baker	Jean	5612130067143	False	English
21	Mr	Bekker	Boris	7710145678083	False	German
19	Mr	Bekker	Hendrik	5607187367892	True	SA
1	Mr	Ferreira	Guiseppe	3712183567065	False	Portugese
9	Mr	Fouche	Jean-Claude	7805263478653	False	French
5	Mrs	Freulich	Ina	8107166893475	False	German
11	Mr	Griffiths	Daffyd	2311078237876	False	Welsh
2	Mrs	Honeywell	Lucy	5907234676012	True	SA
17	Mr	Johnstone	Paul	8002027876086	False	English
4	Mr	Khoza	Bantu	7706161545342	True	SA
6	Mr	Kowalski	Joachim	7407189162348	False	Polish
20	Ms	Kowalski	Nikitha	8012230067083	False	Polish
7	Ms	Krzyaikis	Stavro	7305172654765	False	Greek
3	Ms	Mendes	Isaac	6708139075176	True	SA
13	Ms	Nguni	Priscilla	7609248768934	True	SA
18	Ms	Rodriguez	Laura	8211263788743	False	Spanish
15	Mr	Saint-Ypres	Yves	6610278547278	False	French
16	Mr	Smythe	Andrew	7410287656786	False	English
12	Mrs	Van Rheede	Roxanne	6312054763457	False	Dutch
8	Ms	Von Braun	Gertrude	5608237896212	False	German
14	Mr	Yamimoto	Ishiguru	7907186427868	False	Japanese

(5)

1.2 Complete the code for menu option B (**Mr Ferreira**) by creating a query that will calculate the total amount owed by Mr Ferreira (**ClientNo** field entry is 1). This is a calculated answer. The heading must be named **Total Due** and the result must be formatted to display with TWO decimal places.

Example of output:

Total Due
R47.65

(7)

1.3 All the bookings for the English football fans have been cancelled. Complete the code for menu option C (**English**) by creating a query to delete all English clients (**Nationality** entry field is English) from the **tblClients** table. (4)

1.4 The group uses a 25% markup when calculating the selling price. Complete the code for menu option D (**Cost**) by creating a query that will list the **Date**, **Category, SellingPrice** and **Cost** (selling price minus the 25% markup) for each item ordered by Mr Guiseppe Ferreira (**ClientNo** field entry is 1) from the **tblOrders** table. **Cost** is a calculated field and must be named **Cost**. Use the following formula to calculate the cost:

Cost = selling price – ((selling price/125)*25)

Example of output:

Date	Category	SellingPrice	Cost
2009-01-25	Phone Call	8.75	7.0
2009-01-25	Kitchen	29.95	23.96
2009-01-25	Bar	8.95	7.16

(10)

1.5 The Petersen Group has decided that they want to support all soccer fans by giving them R5,00 discount on the selling price of every item they have ordered if the item's selling price is R30,00 or more. Complete the code for menu option E (**Discount**) by writing a query that will reduce the selling price of all the relevant items by R5,00 in the **tblOrders** table.

NOTE: Every time that you choose the Discount option from the menu the selling prices of items that are still more than R30,00 will be reduced by another R5,00.

Example of output after the Discount option has been executed once:

OrderNo	ClientNo	Date	Category	SellingPrice
1	1	2009-01-25	Phone Call	8.75
2	2	2009-01-25	Bar	84.95
3	3	2009-01-25	Kitchen	44.95
4	1	2009-01-25	Kitchen	29.95
5	1	2009-01-25	Bar	8.95
6	4	2009-01-26	Taxi	154.0
7	5	2009-01-26	Security	25.0
8	5	2009-01-26	Internet	14.95
9	6	2009-01-26	Phone Call	22.86

(5)

1.6 Complete the code for menu option F (**Faltemeyer**) by writing a query that will add the following client data to the **tblClients** table:

Mr Harald Faltemeyer, ID 7407185683074, Swedish

NOTE: The content of the **IDNumber** field is string and the content of the **SA** field is Boolean. (9)

- Enter your examination number as a comment in the first line of the file named **testBandBxxxx.java** containing the SQL statements.
- Save the **testBandBxxxx.java** and the **BandB.java** files.
- A printout of the code for the **testBandBxxxx.java** file will be required. **[40]**

QUESTION 2: JAVA – OBJECT-ORIENTED PROGRAMMING

Guests for the Soccer World Cup want to travel around South Africa to follow their teams. Normally guests have to pay the bill for all the extra items they ordered during their stay when they check out at a guest house. To provide a service which makes them different from (and better than) other B&Bs, the Petersen Group has decided to let their guests transfer their accumulated extra costs between the guest houses in each town. Guests will have to pay the bill for these items when they check out at the last guest house on their journey. They decided that the best way to manage this is to e-mail a text file indicating the extra costs of the guests to the next guest house.

You have been asked to write the program to handle the extra costs of the guests.

The data is stored in a text file named **Extras.txt** in the following format:

GuestNo#GuestName#ExtraType#CostPerItem

An example of some of the data in the text file:

1#Mr G Ferreira#Phone#7.05
2#Mrs L Honeywell#Drinks#71.95
3#Ms I Mendes#Kitchen#39.95
1#Mr G Ferreira#Kitchen#23.95
1#Mr G Ferreira#Drinks#7.15
4#Mr B Khoza#Taxi#127.25
:

Do the following:

- Rename the folder **Question 2 Java** as **Quest2_X** (where X represents your examination number).
- Open this folder and rename the **testExtraItem** file as **testExtraItemxxxx** (where xxxx must be replaced by the last FOUR digits of your examination number).
- Open the **testExtraItemxxxx** file (class) in the folder **Quest2_X.** Change the name of the class to **testExtraItemxxxx** (where xxxx must be replaced by the last FOUR digits of your examination number).
- Execute the program. The following menu will be displayed when you execute the program:

```
        Menu

A - List Items

Q - QUIT

Your choice? :
```

2.1 Create an object class named **ExtraItemxxxx.java** and save this class as **ExtraItemxxxx.java** in your **Quest2_X** folder (where xxxx should be replaced by the last FOUR digits of your examination number.) All fields in this class are private and all methods public. The fields and methods that you need to create and code are described below:

2.1.1 Define a class named **ExtraItemxxxx**. Create appropriately named and typed private fields to hold the following data (suggested field names are given in brackets):

- Guest Number (**guestNum**)
- Item type description (**itemType**)
- Cost per item (**cost**) (10 ÷ 2) (5)

2.1.2 Write a constructor method which accepts the guest number, the item description and the cost per item as parameters. All the fields must be initialised in the constructor. (8 ÷ 2) (4)

2.1.3 Write an appropriately named 'get' method (accessor method) to return the guest number. (4 ÷ 2) (2)

2.1.4 The company uses a 25% markup on cost per item to determine profit. Write a method named **calculateProfit** that calculates and returns the profit (that is, cost*25/100). (4 ÷ 2) (2)

2.1.5 Write a method named **calculatePrice** that calculates the final price of the item (that is, cost + the calculated profit). (4 ÷ 2) (2)

2.1.6 Write a method named **toString** that builds and returns a string with information on the item, formatted as follows:

Item type<tab>Cost<tab>Profit<tab>Final Price

Values must be formatted to two decimal places. (8 ÷ 2) (4)

2.2 Write code to do the following in the **testExtraItemxxxx** file (class):

2.2.1 Create an array named **arrItems** that holds objects of **ExtraItemxxxx**. Write code to read information from the text file **Extras.txt** according to the following steps:

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

(b) Use a loop to ...

- read a line of text from the text file.
- separate the text into the guest number, item type and cost.
- use this information to create a new **ExtraItemxxxx** object for the array named **arrItems**.

(c) Use a counter field to keep track of how many items there are in the array. (28 ÷ 2) (14)

2.2.2 Menu Option A: List Items

When the user selects this menu option the program must do the following:

- Allow the user to enter a guest number
- Search through the array and each time an item for the guest is found …
    - calculate the profit using the percentage mark-up and calculate the final price.
    - display the information using the toString method.
    - add the final price for each item to get a grand total.
- When the search is complete the program must …
    - display the total amount due for the guest.
    - display an appropriate message to say that there are no extra charges for this guest, if no items have been found.

Example of the output:

```
Information on extra items for guest number 1

Item type       Cost            Profit          Price
Phone           R7.05           R1.76           R8.81
Kitchen         R23.95          R5.99           R29.94
Drinks          R7.15           R1.79           R8.94

The total amount due is R47.69
```

(20 ÷ 2) (10)

- Enter your examination number as a comment in the first line of the **testExtraItemxxxx** class as well as the object class (**ExtraItemxxxx**).
- Save all the files.
- Printouts of the code for the two classes (**testExtraItemxxxx** and **ExtraItemxxxx**) will be required.

[43]

QUESTION 3: JAVA PROGRAMMING

You have been given an incomplete program in the folder named **Question 3 Java.**

Do the following:

- Rename the folder named **Question 3 Java** as **Quest3_X**. Replace X with your examination number.
- Open this folder and rename the **testNumbe**r file as **testNumberxxxx** (xxxx should be replaced by the last four digits of your examination number).
- Open the **testNumberxxxx** file in this folder. Change the name of the **testNumbe**r class to **testNumberxxxx** (xxxx should be replaced by the last four digits of your examination number).
- Execute the program. A menu with the following options will be displayed:

```
     MENU

A - Convert
B - Duplicates

Q - QUIT

Your choice? :
```

NOTE: Your solution can have one class with methods OR an object class called **PhoneNumber** containing methods.

The given program generates an array of phone numbers for you to work with. The array called **arrPhoneNos** holds 20 strings. Data in the array will look like this:

086NewHill
086DialBar
086BayView
086KyaSand
086SowetoN
086CasaSol
086TheHavn
086GetFood
086ThaiPlc
086Cleaner
086CasaRok
086RixTaxi
086AirTime
086DialBed
086DialCar
086DialHlp
086KyaRosa
086BaySand
086Cater4U
0861to1Air

3.1 Write code to convert all the alphanumeric characters in the **arrPhoneNos** array into normal telephone numbers and replace them into the same array. Replace the alphabetical characters (upper case or lower case) in the telephone numbers with the corresponding numbers given below:

A, B, C	2
D, E, F	3
G, H, I	4
J, K, L	5
M, N, O	6
P, Q, R, S	7
T, U, V	8
W, X, Y, Z	9

The numeric values in the telephone numbers remain as they are.

NOTE: The resulting numerical telephone number must be reformatted as follows: 3 digits, space, 3 digits, space, 4 digits (for example 086 345 6546).

The program must display the original alphanumeric number and the new numerical version.

NOTE: Use modular programming techniques when solving this problem.

Example of output:

```
Original number Converted number
086NewHill      086 639 4455
086DialBar      086 342 5227
086BayView      086 229 8439
086KyaSand      086 592 7263
086SowetoN      086 769 3866
086CasaSol      086 227 2765
086TheHavn      086 843 4286
086GetFood      086 438 3663
086ThaiPlc      086 842 4752
086Cleaner      086 253 2637
086CasaRok      086 227 2765
086RixTaxi      086 749 8294
086AirTime      086 247 8463
086DialBed      086 342 5233
086DialCar      086 342 5227
086DialHlp      086 342 5457
086KyaRosa      086 592 7672
086BaySand      086 229 7263
086Cater4U      086 228 3748
086ltolAir      086 186 1247
```

(22)

3.2 Use the numerical telephone numbers to check that there are no duplicates in the array. If duplicates are found, the program must display the duplicate numbers. If no duplicates are found, a suitable message must be displayed. At the end of the list there must be a summary stating how many duplicates were found (if any).

Example of the output:

```
Duplicates
086 342 5227
086 227 2765
The number of duplicates : 2
```

Optional: If you wish to further test your program, do the following:

To test for no duplicates, change the given elements of the array as follows:

- Change 086DialBar (the second element in the array) to 086DialBax
- Change 086CasaSol (the sixth element in the array) to 086CasaSox

To test for more than two duplicates, change the given elements of the array as follows:

- Change 086DialBax (the second element in the array) back to 086DialBar
- Change 086CasaSox (the sixth element in the array) back to 086CasaSol
- Replace the last THREE elements of the array with the following:

086AirVine
086BayThe9
086SoxDunn (15)

- Add your examination number as a comment in the first line of the class **testNumbersxxxx**
- Save the class(es).
- A printout of the code for the class **testNumbersxxxx** and any other class(es) you have created will be required. **[37]**

TOTAL SECTION B: 120

GRAND TOTAL: 120

ANNEXURE A: Table Description Sheet

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

tblClients Table Structure

tblClients : Table

Field Name	Data Type	
ClientNo	AutoNumber	Automatically generated Client No
Title	Text	Ms / Mr / Dr etc
Surname	Text	Surname
FName	Text	First name
IDNumber	Text	ID / Passport No
SA	Yes/No	SA citizen (yes / no)
Nationality	Text	Country

tblOrders Table Structure:

tblOrders : Table

Field Name	Data Type	
OrderNo	Number	Unique order number
ClientNo	Number	Number of client who placed the order
Date	Date/Time	Date the order was placed
Category	Text	Category e.g. Bar/ Phone call etc
SellingPrice	Currency	Selling price of item ordered

tblClients Table Data Sample:

tblClients : Table

ClientNo	Title	Surname	FName	IDNumber	SA	Nationality
1	Mr	Ferreira	Guiseppe	3712183567065	☐	Portugese
2	Mrs	Honeywell	Lucy	5907234676012	☑	SA
3	Ms	Mendes	Isaac	6708139075176	☑	SA
4	Mr	Khoza	Bantu	7706161545342	☑	SA
5	Mrs	Freulich	Ina	8107166893475	☐	German
6	Mr	Kowalski	Joachim	7407189162348	☐	Polish
7	Ms	Krzyaikis	Stavro	7305172654765	☐	Greek
8	Ms	Von Braun	Gertrude	5608237896212	☐	German
9	Mr	Fouche	Jean-Claude	7805263478653	☐	French
10	Ms	Baker	Jean	5612130067143	☐	English
11	Mr	Griffiths	Daffyd	2311078237876	☐	Welsh
12	Mrs	Van Rheede	Roxanne	6312054763457	☐	Dutch
13	Ms	Nguni	Priscilla	7609248768934	☑	SA
14	Mr	Yamimoto	Ishiguru	7907186427868	☐	Japanese
15	Mr	Saint-Ypres	Yves	6610278547278	☐	French
16	Mr	Smythe	Andrew	7410287656786	☐	English
17	Mr	Johnstone	Paul	8002027876086	☐	English
18	Ms	Rodriguez	Laura	8211263788743	☐	Spanish
19	Mr	Bekker	Hendrik	5607187367892	☑	SA
20	Ms	Kowalski	Nikitha	8012230067083	☐	Polish
21	Mr	Bekker	Boris	7710145678083	☐	German

tblOrders Table Data Sample:

tblOrders : Table

OrderNo	ClientNo	Date	Category	SellingPrice
1	1	25/01/2009	Phone Call	R 8.75
2	2	25/01/2009	Bar	R 89.95
3	3	25/01/2009	Kitchen	R 49.95
4	1	25/01/2009	Kitchen	R 29.95
5	1	25/01/2009	Bar	R 8.95
6	4	26/01/2009	Taxi	R 159.00
7	5	26/01/2009	Security	R 30.00
8	5	26/01/2009	Internet	R 14.95
9	6	26/01/2009	Phone Call	R 22.86

ANNEXURE B: Instruction to create the database BandB.mdb

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

- Use the two text files named **tblClients** and **tblOrders** supplied. Create your own database named BandB.mdb with a table named **tblClients** and another table named **tblOrders** in the **Question 1** folder.
- Change the data types and the sizes of the fields in the two tables to the specifications given below.
- Create a relationship (with referential integrity) between the **tblClients** table and the **tblOrders** table.

The fields in the **tblClients** table are defined as follows:

Field Name	Type	Size	Comment
ClientNo	AutoNumber		Unique code of the client
Title	Text	4	Mr. / Mrs. /Dr etc.
Surname	Text	20	Surname
FName	Text	20	First name
IDNumber	Text	20	ID / Passport number
SA	Yes/No		SA Citizen
Nationality	Text	20	Country

See ANNEXURE A: Example of the data contained in the **tblClients** table.

Content of the tblClient.txt file:

```
"ClientNo","Title","Surname","FName","IDNumber","SA","Nationality"
1,"Mr","Ferreira","Guiseppe","3712183567065",0,"Portugese"
2,"Mrs","Honeywell","Lucy","5907234676012",1,"SA"
3,"Ms","Mendes","Isaac","6708139075176",1,"SA"
4,"Mr","Khoza","Bantu","7706161545342",1,"SA"
5,"Mrs","Freulich","Ina","8107166893475",0,"German"
6,"Mr","Kowalski","Joachim","7407189162348",0,"Polish"
7,"Ms","Krzyaikis","Stavro","7305172654765",0,"Greek"
8,"Ms","Von Braun","Gertrude","5608237896212",0,"German"
9,"Mr","Fouche","Jean-Claude","7805263478653",0,"French"
10,"Ms","Baker","Jean","5612130067143",0,"English"
11,"Mr","Griffiths","Daffyd","2311078237876",0,"Welsh"
12,"Mrs","Van Rheede","Roxanne","6312054763457",0,"Dutch"
13,"Ms","Nguni","Priscilla","7609248768934",1,"SA"
14,"Mr","Yamimoto","Ishiguru","7907186427868",0,"Japanese"
15,"Mr","Saint-Ypres","Yves","6610278547278",0,"French"
16,"Mr","Smythe","Andrew","7410287656786",0,"English"
17,"Mr","Johnstone","Paul","8002027876086",0,"English"
18,"Ms","Rodriguez","Laura","8211263788743",0,"Spanish"
19,"Mr","Bekker","Hendrik","5607187367892",1,"SA"
20,"Ms","Kowalski","Nikitha","8012230067083",0,"Polish"
21,"Mr","Bekker","Boris","7710145678083",0,"German"
```

ANNEXURE B (continued)

The fields in the **tblOrders** table are defined as follows:

Field Name	Type	Size	Comment
OrderNo	Number	Byte	Unique code of Order
ClientNo	Number	Longint	Number of client who placed the order
Date	Date/Time	ShortDate	Date the order was placed
Category	Text	20	
SellingPrice	Currency		Selling price of the item ordered

See ANNEXURE A: Example of the data contained in the **tblOrders** table.

Content of the tblOrders.txt file:

```
"OrderNo","ClientNo","Date","Category","SellingPrice"
1,1,25/1/2009 0:00:00,"Phone Call",R8.75
2,2,25/1/2009 0:00:00,"Bar",R89.95
3,3,25/1/2009 0:00:00,"Kitchen",R49.95
4,1,25/1/2009 0:00:00,"Kitchen",R29.95
5,1,25/1/2009 0:00:00,"Bar",R8.95
6,4,26/1/2009 0:00:00,"Taxi",R159.00
7,5,26/1/2009 0:00:00,"Security",R30.00
8,5,26/1/2009 0:00:00,"Internet",R14.95
9,6,26/1/2009 0:00:00,"Phone Call",R22.86
```

ANNEXURE C: Instruction to connect to the database in Delphi

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

- Click on the ADOQuery component named **qryBandB**.
- Click on the Ellipse button (three dots) to the right of the Connection string property in the Object Inspector.
- Click on the Build button which takes you to the Data Link Properties dialog box.
- Select Microsoft Jet 4.0 OLE DB Provider and click on Next.
- The first option on the Connection tab sheet allows you to browse and find the **BandB.mdb** file.
- Remove the user name Admin.
- Click on the Test Connection button.
- Click OK on each one of the open dialog windows.