education

Department:
Education
REPUBLIC OF SOUTH AFRICA

NATIONAL SENIOR CERTIFICATE

GRADE 12

INFORMATION TECHNOLOGY P1 (Practical)

FREE STATE

COMMON PAPER JUNE 2009

MARKS: 120

TIME: 3 hours

This question paper consists of 7 pages.

INSTRUCTIONS AND INFORMATION

1. This is a three hour examination.

2. Answer all the questions.

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/educator will tell you where to find them on the hard drive of the workstation you are using or in a network folder.

Question 1	**WorldCup.mdb**
Question 2	**stadiums.txt**
Question 3	**seats.dat**

4. If a disk or CD containing the above files was issued to you, write your name on the label.

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

6. Save ALL your solutions in folders with the number of the question and your initials and surname as the name of the folder, for example Quest2_JSmith.

7. Type in your name and surname as a comment in the first line of each program.

8. Read ALL the questions carefully. Do only what is required.

9. At the end of this examination session you will be required to hand in the disk or CD with all the files with the work you have done or you must make sure that ALL the files with your work have been saved on the network as explained to you by the invigilator/educator. Ensure that ALL files can be read.

10. During the examination you may use the HELP functions of the software. You may NOT refer to any other resource material.

11. Make printouts of the programming code of ALL the questions. Make sure that your name and surname appears in the first line of each program as a comment. Arrange the pages of each question in the correct order and then the questions from QUESTION 1 to 3. Staple all the printouts of the questions (arranged correctly) in one batch to hand in.

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

Scenario:
In preparation of the 2010 Soccer World Cup you have been approached to develop concept programs to be used throughout the competition.

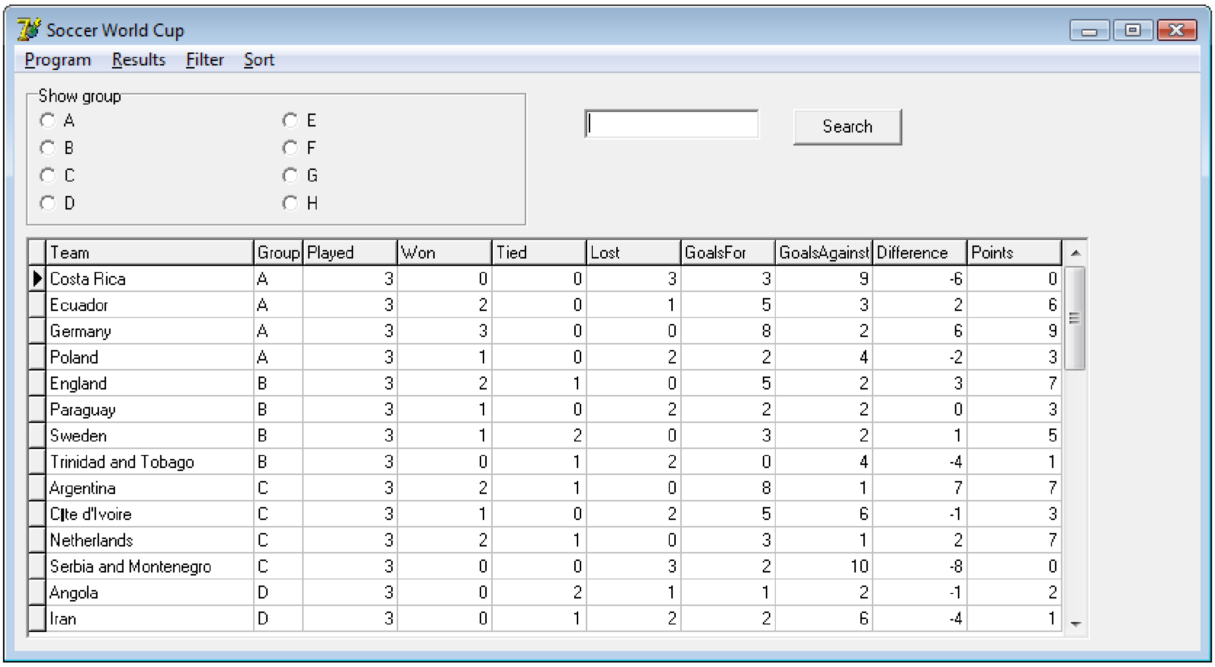
QUESTION 1 : DELPHI PROGRAMMING AND DATABASE

The provided MS Access file (WorldCup.mdb) contains a set of test data from possible games by participating countries – this is based on the results of the previous soccer world cup.

Take note that the fields refer to the following:

Team	Country
Group	Group the country belongs to
Played	Games played
Won	Games won
Tied	Games tied
Lost	Games lost
GoalsFor	Goals gained by this team
GoalsAgainst	Goals gained against this team
Difference	GoalsFor minus GoalsAgainst
Points	Points received

1.1 Create a program in Delphi that includes an Edit box (edtName), a Button (btnSearch), a RadioGroup (rgpGroups) as well as a MainMenu component (mnuSoccer): (2)



Team	Group	Played	Won	Tied	Lost	GoalsFor	GoalsAgainst	Difference	Points
Costa Rica	A	3	0	0	3	3	9	-6	0
Ecuador	A	3	2	0	1	5	3	2	6
Germany	A	3	3	0	0	8	2	6	9
Poland	A	3	1	0	2	2	4	-2	3
England	B	3	2	1	0	5	2	3	7
Paraguay	B	3	1	0	2	2	2	0	3
Sweden	B	3	1	2	0	3	2	1	5
Trinidad and Tobago	B	3	0	1	2	0	4	-4	1
Argentina	C	3	2	1	0	8	1	7	7
Clte d'Ivoire	C	3	1	0	2	5	6	-1	3
Netherlands	C	3	2	1	0	3	1	2	7
Serbia and Montenegro	C	3	0	0	3	2	10	-8	0
Angola	D	3	0	2	1	1	2	-1	2
Iran	D	3	0	1	2	2	6	-4	1

*(Save the unit as **Question1u.pas** and the project file as **Question1p.dpr**)*

1.2 The MainMenu (mnuSoccer) must include the following items: (2)

Program	**Results**	**Filter**	**Sort**
Exit	Show All	Teams with wins	Name
	Get Difference	Positive Difference	Games Won
	Zero Difference		Points
	Win Point		

1.3 The RadioGroup (rgpGroups) should include the letters A through to H as items. (2)

1.4 Add an ADOQuery (qryResults), DataSource (datSoccer) and DBGrid (dbgOutput). (3)

Set the connection by following these steps:

- Click on the ADOQuery component named qryResults
- 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 WorldCup.mdb file
- Remove the user name Admin
- Click on the Test Connection button
- Click OK on each one of the open dialog windows

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

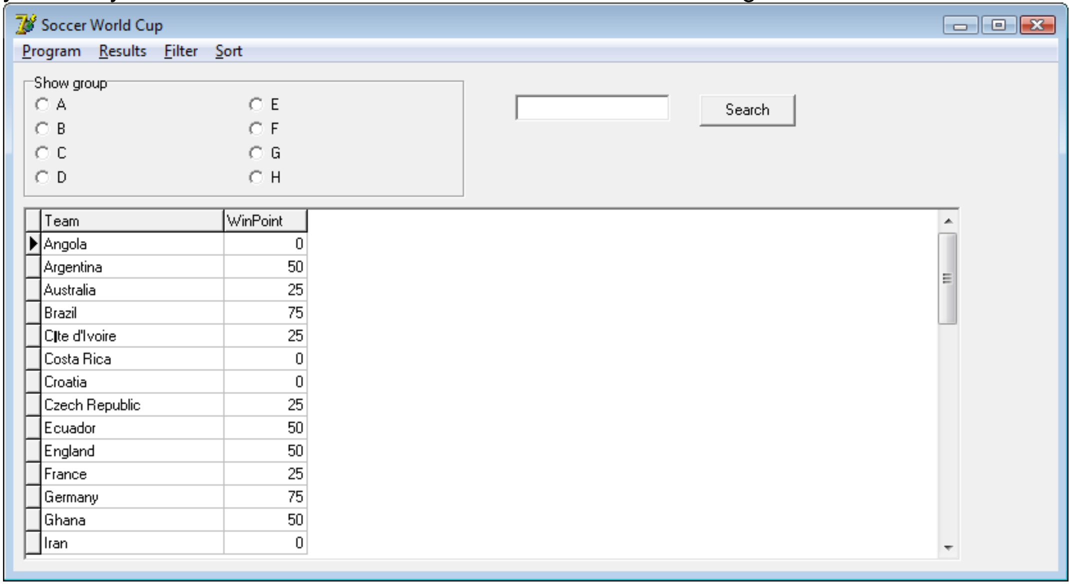
1.5 Complete the coding for the Program > Exit menu option. This option should allow the user to close the program. (1)

1.6 Set up the following SQL queries for the menu items in mnuSoccer:
1.6.1 Results > Show All: show all the fields and records. (2)

1.6.2 Results > Get Difference: Calculate the difference between GoalsFor and GoalsAgainst and place the result in the Difference field. Then display all the fields. (3)

1.6.3 Results > Zero Difference: Set the Difference field of all the records to 0. Then display all the fields. (2)

1.6.4 Results > Win Point: Create a new field WinPoint calculated multiplying the amount of games won by 25. Only show the Team name and WinPoint sorted according to the team name. (3)



Team	WinPoint
Angola	0
Argentina	50
Australia	25
Brazil	75
C▮te d'Ivoire	25
Costa Rica	0
Croatia	0
Czech Republic	25
Ecuador	50
England	50
France	25
Germany	75
Ghana	50
Iran	0

1.6.5 Filter > Teams with wins: only show teams that won games (show all fields). (2)

1.6.6 Filter > Positive Difference: only show teams with a zero or positive difference (show all fields). (2)

1.6.7 Sort > Name: display all the records and sort them according to the team name. (2)

1.6.8 Sort > Games Won: display all the records and sort them according to the games won (least first). (2)

1.6.9 Sort > Points: display all the records and sort them according to the points (highest first). (2)

1.7 Use the Edit box (edtName) and the Button (btnSearch) to only show the teams with a team name as specified in the Edit box. (Show all fields.) (4)

1.8 Use the RadioGroup (rgpGroups) to only show the teams that belong to a certain Group – from A to H. (Show all fields.) (6)

Save all files. Print out all coding as well as a screen shot of the form.

[40]

QUESTION 2 : DELPHI PROGRAMMING (OOP)

Object-oriented programming skills will be tested with this question. You are required to produce a solution that includes the class specified in the instructions. No marks will be allocated to any alternative solution such as a program not creating an object.

The text file **stadiums.txt** includes the following data:

CITY	STADIUM	SEATS	CAT. 1	CAT. 2	CAT. 3	CAT. 4
Johannesburg	Soccer City	94700	15	20	25	40
Durban	Moses Mabhida Stadium	70000	23	27	25	25
Cape Town	Green Point Stadium	69070	10	20	30	40
Johannesburg	Coca-Cola Park	62500	10	23	34	33
Pretoria	Loftus Versfeld Stadium	51760	10	15	35	40
Port Elizabeth	Nelson Mandela Bay Stadium	48000	10	10	40	40
Bloemfontein	Free State Stadium	48000	1	32	33	34
Polokwane	Peter Mokaba Stadium	46000	25	25	25	25
Nelspruit	Mbombela Stadium	44000	0	25	25	50
Rustenburg	Royal Bafokeng Stadium	42000	5	21	32	42

Each column is separated with a comma in the file, e.g. Johannesburg,Soccer City,94700,15,20,25,40

The file includes the city where the stadium can be found, the name of the stadium and the total number of seats. The last four category columns refer to the percentage of seats allocated to each different type of ticket category.

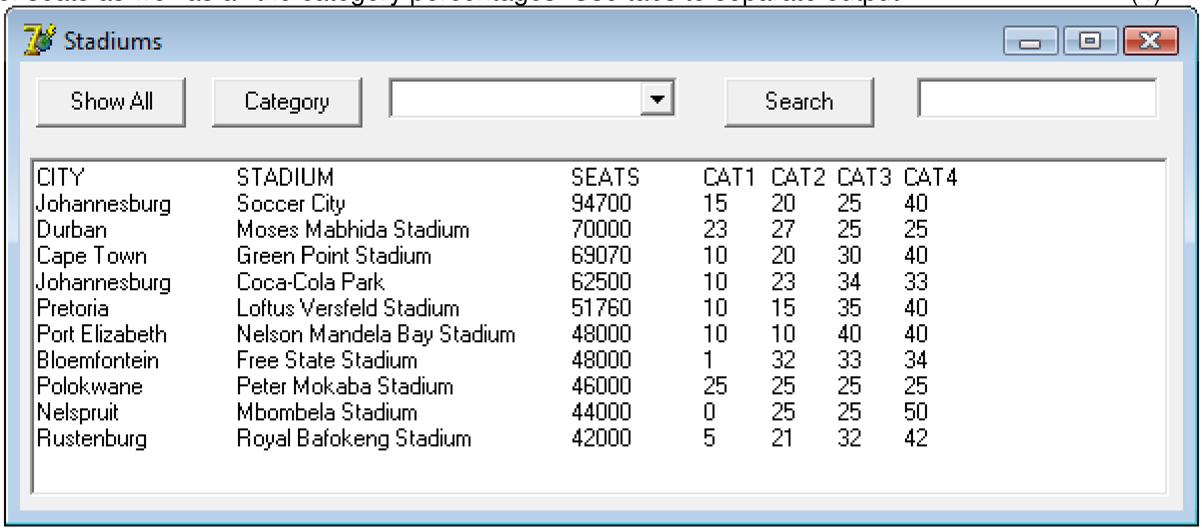
2.1 An object class (***StadiumUnit***) should be created and saved in the same folder as the main unit (***Question2u***) and the project file (***Question2p***). Add the following code to your own class:

2.1.1 Define a class named **TStadium**. The following private fields should be used: (2)

```
fCity : String;
fName : String;
fSeats : Integer;
fCategory1 : Integer;
fCategory2 : Integer;
fCategory3 : Integer;
fCategory4 : Integer;
```

2.1.2 A public parameterised constructor named **create** that will pass values for the fields in the class must be created. These parameters will initialise the fields of the class. (4)

2.1.3 A public method (**getStadium**) should be written to return the city name, stadium name, number of seats as well as all the category percentages. Use tabs to separate output. (4)



```
CITY            STADIUM                      SEATS      CAT1  CAT2  CAT3  CAT4
Johannesburg    Soccer City                  94700      15    20    25    40
Durban          Moses Mabhida Stadium        70000      23    27    25    25
Cape Town       Green Point Stadium          69070      10    20    30    40
Johannesburg    Coca-Cola Park               62500      10    23    34    33
Pretoria        Loftus Versfeld Stadium      51760      10    15    35    40
Port Elizabeth  Nelson Mandela Bay Stadium   48000      10    10    40    40
Bloemfontein    Free State Stadium           48000      1     32    33    34
Polokwane       Peter Mokaba Stadium         46000      25    25    25    25
Nelspruit       Mbombela Stadium             44000      0     25    25    50
Rustenburg      Royal Bafokeng Stadium       42000      5     21    32    42
```

2.1.4 A public method (**getCity**) should be written to return only the city name. (1)

2.1.5 Public methods should be created for each of the categories (**getCat1**, **getCat2**, **getCat3** and **getCat4**). These methods should return the name of the stadium as well as the number of seats in terms of the percentage and the total seats available per stadium. Furthermore it should also return the total income to be made from that particular category and stadium. (Category 1: R1120; category 2: R840; category 3: R560; and category 4: R140.) (8)

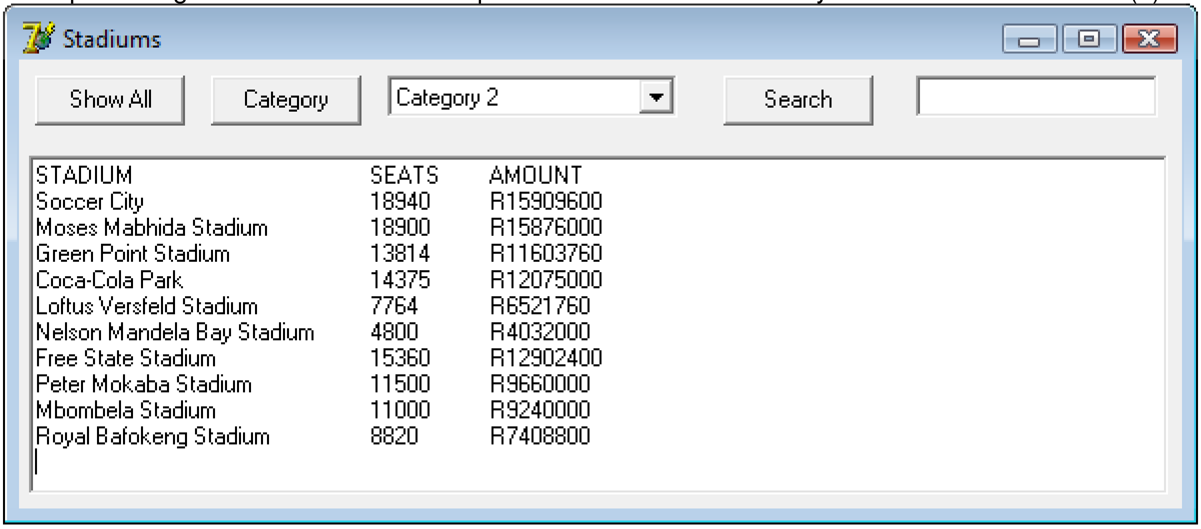
2.2 Complete the main unit (Question2U):

2.2.1 Create an array called **arrStadiums** for objects of **TStadium**. Add the following code in the OnActivate event handler:

2.2.1.1 Assign textfile stadiums.txt to TextFile variable fFile. (2)
2.2.1.2 Use counter (iCount) to count items in the array. (1)
2.2.1.3 Isolate the city name, stadium name, number of seats and four category percentages and store them in variables. (4)
2.2.1.4 Create a new TStadium object and place it in the array arrStadiums. (2)

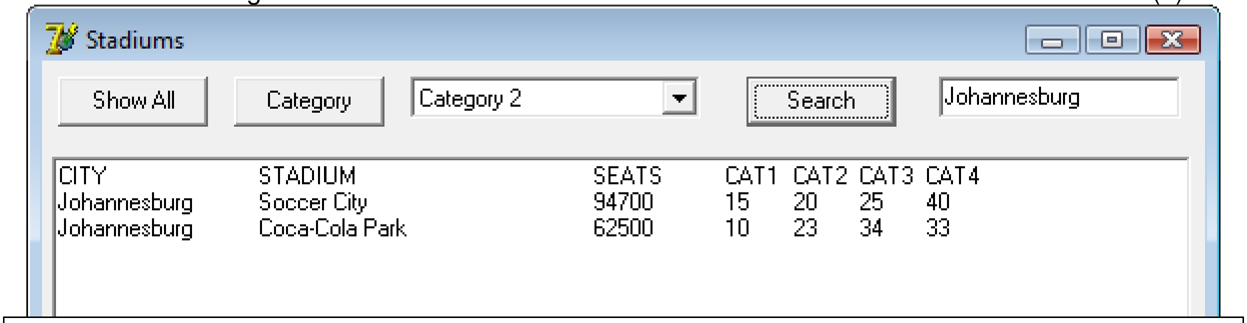
2.2.2 Add a Button (btnShowAll) that sets appropriate RichEdit (redOutput) tabs, headings and invokes the getStadium method. (3)

2.2.3 Add another Button (btnCategory) and a ComboBox (cmbCategory) with which a category can be chosen and the relevant GetCat1, GetCat2, GetCat3 or GetCat4 be invoked. Set appropriate RichEdit tabs, headings and display the name of the stadium as well as the number of seats in terms of the percentage and the total available per stadium as determined by the method invoked. (4)



```
STADIUM                     SEATS    AMOUNT
Soccer City                 18940    R15909600
Moses Mabhida Stadium       18900    R15876000
Green Point Stadium         13814    R11603760
Coca-Cola Park              14375    R12075000
Loftus Versfeld Stadium     7764     R6521760
Nelson Mandela Bay Stadium  4800     R4032000
Free State Stadium          15360    R12902400
Peter Mokaba Stadium        11500    R9660000
Mbombela Stadium            11000    R9240000
Royal Bafokeng Stadium      8820     R7408800
```

2.2.4 Add another Button (btnSearch) and an Edit box (edtCity) that can be used in conjunction with the GetCity method to search for a specific city and then display the city name, stadium name, number of seats as well as all the category percentages as determined by the GetStadium method. Set tabs and add the headings shown below. (5)



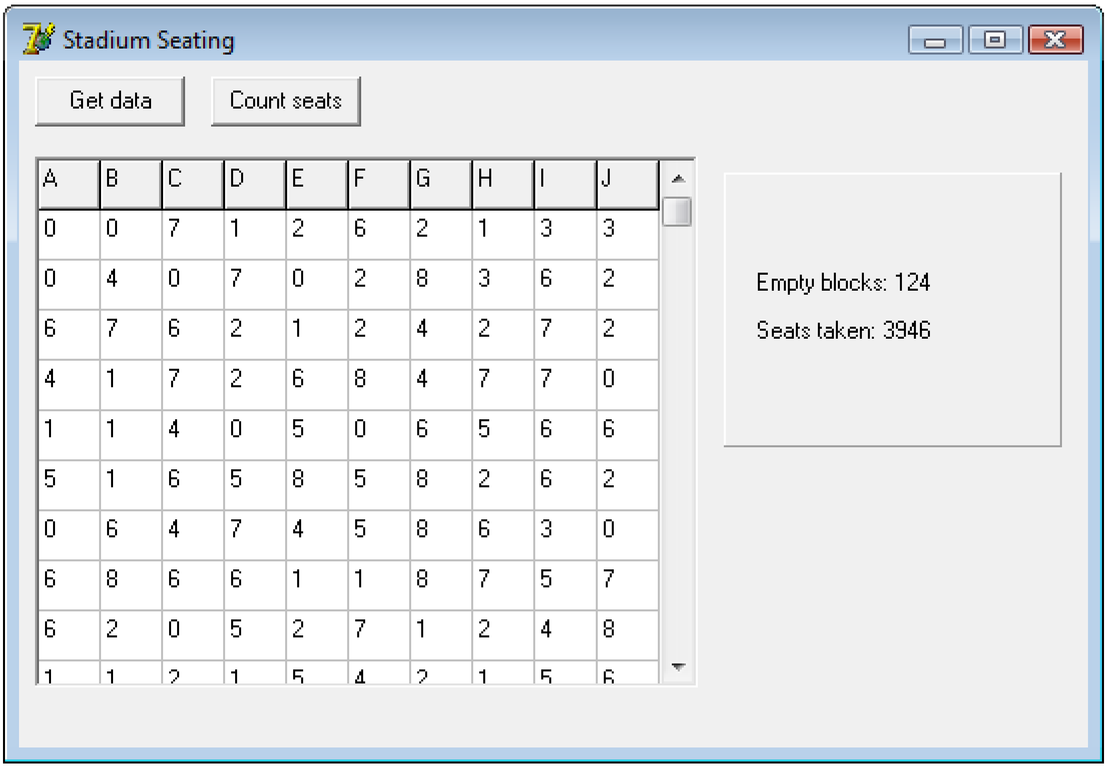
```
CITY          STADIUM                   SEATS     CAT1  CAT2  CAT3  CAT4
Johannesburg  Soccer City               94700     15    20    25    40
Johannesburg  Coca-Cola Park            62500     10    23    34    33
```

Save all files. Print out all coding as well as screen shots.

[40]

QUESTION 3 : DELPHI PROGRAMMING

You have been asked to design an interface to be used to track seats allocated to a section of a stadium. Each cell in the StringGrid (sgdSeats) represents a block of 8 possible seats. Use the screen shot below as a guideline for your program.



A	B	C	D	E	F	G	H	I	J
0	0	7	1	2	6	2	1	3	3
0	4	0	7	0	2	8	3	6	2
6	7	6	2	1	2	4	2	7	2
4	1	7	2	6	8	4	7	7	0
1	1	4	0	5	0	6	5	6	6
5	1	6	5	8	5	8	2	6	2
0	6	4	7	4	5	8	6	3	0
6	8	6	6	1	1	8	7	5	7
6	2	0	5	2	7	1	2	4	8
1	1	2	1	5	4	2	1	5	6

Empty blocks: 124

Seats taken: 3946

(Save the unit as ***Question3u.pas*** *and the project file as* ***Question3p.dpr****)*

3.1 Create a program with the following components on the Form (frmSeating): (6)

Component	Name	Properties
Button	btnGetData	Caption: Get data
Button	btnCount	Caption: Count seats
StringGrid	sgdSeats	Columns: 10 Rows: 101 Default row and column size: 30 One fixed row
Panel	pnlResult	Caption empty
Label	lblEmpty	Caption: Empty blocks
Label	lblSeats	Caption: Seats taken

3.2 Add the letters A to J in the first (fixed) row of the StringGrid (sgdSeats). (1)

3.3 Use the first Button (btnGetData) to load the data from the text file **seats.dat**. Each line of the text file contains 10 numbers that should be separated into one digit per cell. (15)

3.4 Use the second Button (btnCount) to determine how many empty cells (with a value of 0) are available. Display this in Label *lblEmpty* – see screenshot above. Also get the total of all the seats taken by adding all the numbers in the StringGrid together and then display it in Label *lblSeats*. (18)

Save all files. Print out all coding as well as a screen shot.

[40]

TOTAL: 120