



# education

---

Department:  
Education  
**REPUBLIC OF SOUTH AFRICA**

**NATIONAL  
SENIOR CERTIFICATE**

**GRADE 12**

**INFORMATION TECHNOLOGY P1**

**FEBRUARY/MARCH 2009**

**MARKS: 120**

**TIME: 3 hours**

**This question paper consists of 30 pages.**

**INSTRUCTIONS AND INFORMATION**

1. This is a three-hour examination. 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:

**QUESTION 1****Delphi:**

AdvertsTb.txt  
ClientsTb.txt  
NewspaperDB.mdb  
NewspaperDBase\_U.pas  
NewspaperDBase\_P.dpr  
NewspaperDBase\_U.dfm

**Java:**

AdvertsTb.txt  
ClientsTb.txt  
NewspaperDB.mdb  
testNewsPaper.java  
NewsPaper.java

**QUESTION 2****Delphi:**

Advertisements.txt  
testAdverts\_U.pas  
testAdverts\_U.dfm  
testAdverts\_P.dpr

**Java:**

Advertisements.txt  
testAdverts.java

4. If a disk or CD containing the above files was issued to you, write your name and examination number 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 examination number as the name of the folder, for example Quest2\_3020160012
7. Type in your examination number as a comment in the first line of each program.
8. Read ALL the questions carefully. Do NOT do more than is required by the question.
9. At the end of this examination session you will be required to hand in the disk or CD given to you by the invigilator with all your work saved on it OR you must make sure that all your work have been saved on the network as explained to you by the invigilator/teacher. Ensure that all files can be read.

10. 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.
11. All printing of programming questions will take place within an hour of the completion of the examination.

## SECTION A

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

### QUESTION 1: DELPHI – PROGRAMMING AND DATABASE

The advertising department of the local newspaper has a classified section where readers can place advertisements on a weekly basis. They decided to make use of a database to assist them in managing the placement of these advertisements. A program is being developed to make it easier for the staff at the newspaper to answer certain queries being made frequently. You are required to assist in developing this program.

The information on the clients who placed advertisements as well as the advertisements being placed have been captured in a database called **NewspaperDB.mdb**.

The database, **NewspaperDB.mdb**, which contains two tables named **ClientsTb** and **AdvertsTb** respectively, has been supplied to you in a folder named **Question 1 Delphi**.

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

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

The **ClientsTb** table stores data on the clients of the newspaper. The fields in the **ClientsTb** table are defined as follows:

<u>Field Name</u>	<u>Type</u>	<u>Size</u>	<u>Comment</u>
ClientID	Text	5	Unique code of the client
ClientName	Text	30	
TelNum	Text	15	Contact number of the client
Suburb	Text	30	Suburb where the client lives

The following table (see next page) is an example of the data contained in the **ClientsTb** table:

	<b>ClientID</b>	<b>ClientName</b>	<b>TelNum</b>	<b>Suburb</b>
+	BE013	Best Cars	(011) 678 9883	Bedfordview
+	CA012	CarSelect	(011) 098 3445	Roodepoort
+	DB002	DogBox	082 123 3210	Midrand
+	DO007	DOE	(011) 285 9887	Bedfordview
+	EC009	EconoTravel	(011) 542 3438	Johannesburg
+	HI004	HiTech Goodies	(011)567 2341	Roodepoort
+	JF008	Job For You	074 326 7851	Soweto
+	PE014	Pedigree Cars	(012) 343 9886	Centurion
+	PH006	PhotoPro	072 786 9880	Centurion
+	PO003	PowerBackup Inc	(012)987 6543	Pretoria Noord
+	PP001	The Pet Place	083 456 7891	Centurion
+	PR010	Property Wizard	084 009 8743	Pretoria
+	SC005	Scappers Heaven	(012) 609 8765	Pretoria Noord
+	SP011	SportsPro	(012) 025 6331	Centurion

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

<b><u>Field Name</u></b>	<b><u>Type</u></b>	<b><u>Size</u></b>	<b><u>Comment</u></b>
AdvertID	Text	5	Unique code of the advert
ClientID	Text	5	Code of the client
DatePlaced	Date/Time	ShortDate	
WeeksRunning	Number	Integer	
NumWords	Number	Integer	Related to the advert placed
Paid	Boolean		Yes if paid. No if not paid
Category	Text	20	Category where advert will be placed

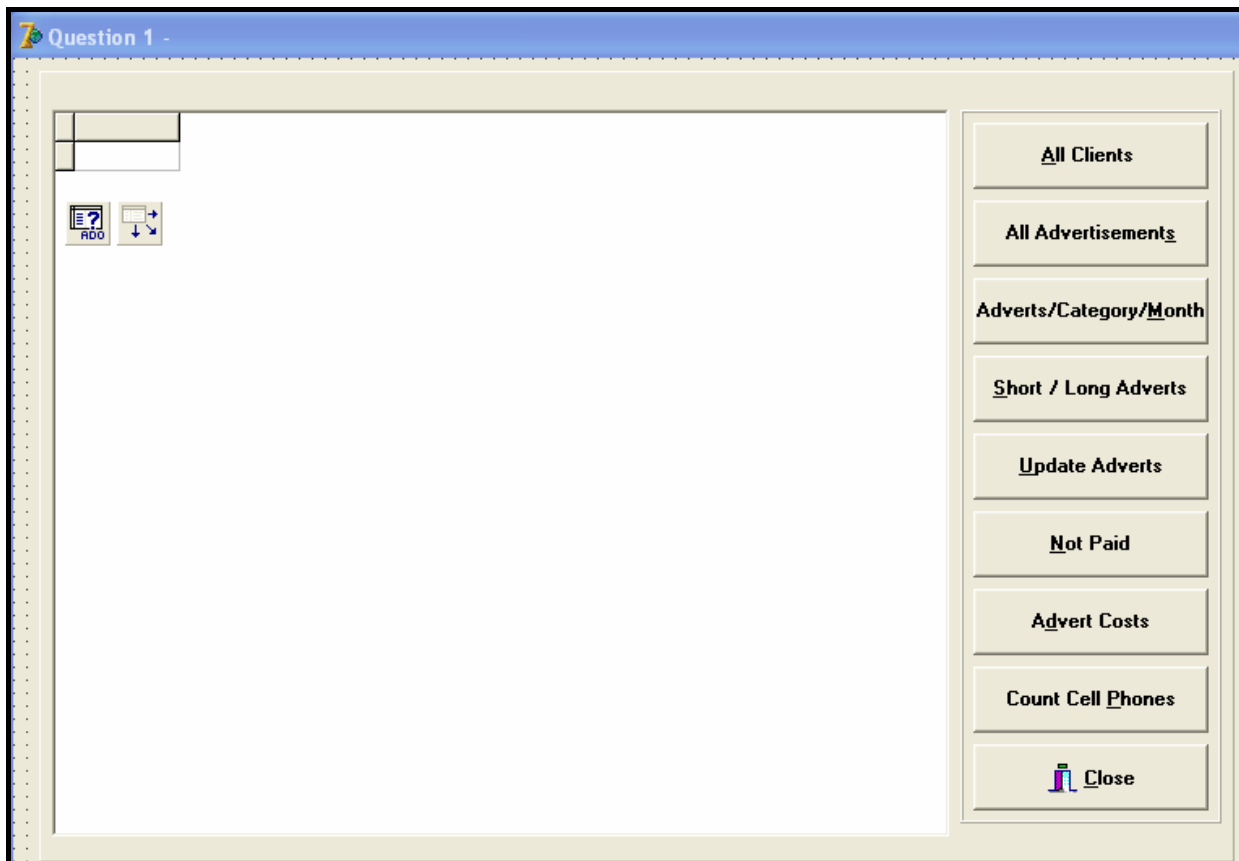
The following table (see next page) is an example of the data contained in the **AdvertsTb** table:

AdvertID	ClientID	DatePlaced	WeeksRunning	NumWords	Paid	Category
AN001	PP001	2008/10/19	2	15	<input checked="" type="checkbox"/>	Animals
AN002	DB002	2008/10/28	3	27	<input type="checkbox"/>	Animals
CA001	BE013	2008/12/10	3	15	<input checked="" type="checkbox"/>	Cars
CA002	BE013	2008/10/10	3	24	<input type="checkbox"/>	Cars
CA003	CA012	2008/10/24	2	14	<input type="checkbox"/>	Cars
CA004	CA012	2008/11/12	2	15	<input type="checkbox"/>	Cars
CA005	PE014	2008/12/15	3	35	<input type="checkbox"/>	Cars
CA006	BE013	2008/12/01	2	25	<input checked="" type="checkbox"/>	Cars
CO001	HI004	2008/11/23	2	12	<input checked="" type="checkbox"/>	Computers
CO002	HI004	2008/11/20	2	16	<input checked="" type="checkbox"/>	Computers
CO003	HI004	2008/11/16	3	14	<input checked="" type="checkbox"/>	Computers
CO004	HI004	2008/10/12	2	10	<input type="checkbox"/>	Computers
JO001	DO007	2008/10/12	1	35	<input checked="" type="checkbox"/>	Jobs
JO002	DO007	2008/11/01	1	30	<input checked="" type="checkbox"/>	Jobs
PH001	PH006	2008/12/10	1	30	<input checked="" type="checkbox"/>	Photographic
PR001	PR010	2008/12/10	2	15	<input checked="" type="checkbox"/>	Properties
PR002	PR010	2008/10/28	1	30	<input checked="" type="checkbox"/>	Properties
PR003	PR010	2008/11/08	2	24	<input type="checkbox"/>	Properties
SP001	SP011	2008/10/10	2	18	<input type="checkbox"/>	Sports
SP002	SP011	2008/11/12	2	20	<input checked="" type="checkbox"/>	Sports
SP003	SP011	2008/12/10	2	18	<input checked="" type="checkbox"/>	Sports
ST001	SC005	2008/12/11	1	35	<input type="checkbox"/>	Stationery
ST002	SC005	2008/11/26	2	12	<input type="checkbox"/>	Stationery

You have also been supplied with an incomplete Delphi program with a unit named **NewspaperDBase\_U** and a project named **NewspaperDBase\_P** in the **Question 1 Delphi** folder. Open the incomplete program. It displays nine buttons as well as a DBGrid that will be used as an output component.

Do the following:

- Add your examination number to the caption of the form to the right of 'Question 1 \_!'.
- Change the **captions** of the buttons from top to bottom to match the screenshot (FIGURE 1.1) (see next page).



**FIGURE 1.1**

Change the names of the buttons from top to bottom as follows: **btnDisplayClients**, **btnAllAdverts**, **btnCatMonth**, **btnShortLongAdverts**, **btnUpdateAdverts**, **btnNotPaid**, **btnCostAdverts** and **btnCountCellphones** respectively as indicated in QUESTIONS 1.1 to 1.8.

- The program should be able to connect to the **NewspaperDB.mdb** database. When you do QUESTION 1.1 (which follows on the next page) and you find that the connectivity is not in place, use the following steps to establish the connection with the database:
  - Click on the ADOQuery component called **qryAdverts**.
  - 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 dialogue 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 **NewspaperDB.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 and submit the programming code for marking.

**Marks will only be awarded for the program methods that contain the SQL statements in the unit named NewsPaperDBase\_U.**

Complete the program by creating the necessary SQL statements in the buttons named **btnDisplayClients**, **btnAllAdverts**, **btnCatMonth**, **btnShortLongAdverts**, **btnUpdateAdverts**, **btnNotPaid**, **btnCostAdverts** and **btnCountCellphones** respectively as indicated in QUESTIONS 1.1 to 1.8.

1.1 Complete the code in the **DisplayClients** button by formulating an SQL statement to display all the fields in the **ClientsTb** table in alphabetical order according to the names of the clients. Example of output:

ClientID	ClientName	TelNum	Suburb
BE013	Best Cars	(011) 678 9883	Bedfordview
CA012	CarSelect	(011) 098 3445	Roodepoort
DO007	DOE	(011) 285 9887	Bedfordview
DB002	DogBox	082 123 3210	Midrand
EC009	EconoTravel	(011) 542 3438	Johannesburg

:  
:

(3)

1.2 Complete the code in the **AllAdverts** button by formulating an SQL statement to display all the fields in the **AdvertsTb** table except the **ClientID** field. Also display the **ClientName** field from the **ClientsTb** table. NOTE: You will need to link the tables with an appropriate **where/join** clause to be able to do this. Example of the output:

AdvertID	DatePlaced	WeeksRunning	NumWords	Paid	Category	ClientName
AN001	2008/10/19	2	12	True	Animals	The Pet Place
AN002	2008/10/28	3	27	False	Animals	DogBox
CO002	2008/11/20	2	16	True	Computers	HiTech Goodies
CO004	2008/10/12	2	10	False	Computers	HiTech Goodies
CO001	2008/11/23	2	12	True	Computers	HiTech Goodies
CO003	2008/11/16	3	14	True	Computers	HiTech Goodies
ST002	2008/11/26	2	12	False	Stationary	Scappers Heaven
ST001	2008/12/11	1	35	False	Stationery	Scappers Heaven
PH001	2008/12/10	1	30	True	Photographic	PhotoPro

:  
:

(3)

1.3 Complete the code in the **CatMonth** button, by asking the user to enter the name of the category and the number of the month as inputs. Formulate an SQL statement to display the **AdvertID**, **DatePlaced** and **WeeksRunning** fields for the adverts in the category and month entered by the user. Example of the output where the user enters 'Computers' as the category and 11 as the number of the month (on the next page):



AdvertID	DatePlaced	WeeksRunning
CO002	2008/11/20	2
CO001	2008/11/23	2
CO003	2008/11/16	3

(6)

- 1.4 Complete the code in the **ShortLongAdverts** button by formulating an SQL statement to display the **AdvertID**, the number of words and the names of the clients who placed adverts with less than 15 and more than 30 words. Display suitable headings. NOTE: You will need to link the tables with an appropriate **where/join** clause to be able to do this. Example of the output:

AdvertID	Number of Words	Client Name
AN001	12	The Pet Place
CO004	10	HiTech Goodies
CO001	12	HiTech Goodies
CO003	14	HiTech Goodies
ST002	12	Scappers Heaven
ST001	35	Scappers Heaven
JO001	35	DOE
CA003	14	CarSelect
CA005	35	Pedigree Cars

(6)

- 1.5 Complete the code in the **UpdateAdverts** button by asking the user to enter the ID of the advert and the number of words to be added. Formulate an SQL statement to update the number of words for the advert and then display the IDs and the number of words of all the advertisements after it has been updated.

(6)

- 1.6 Complete the code in the **NotPaid** button by asking the user to enter a letter of the alphabet. Formulate an SQL statement to display the IDs of the adverts that have not been paid for in the categories starting with the letter entered by the user. Also display the names of the categories and the names of the clients as well as the **Paid** field. NOTE: You will need to link the tables with an appropriate **where/join** clause to be able to do this. Example of the output if the user enters the letter S as input:

AdvertID	Category	Name of Client	Has Paid
ST002	Stationery	Scappers Heaven	False
SP001	Sports	SportsPro	False
ST001	Stationery	Scappers Heaven	False

(6)

1.7 Complete the code in the **CostAdverts** button. Formulate an SQL statement to list the **AdvertID**, **DatePlaced**, **NumWords**, **WeeksRunning** and **Cost** of all the advertisements placed after 16 November 2008. **Cost** is a calculated field. The cost of an advertisement must be calculated at 50c per word per week. Example of output:

AdvertID	DatePlaced	NumWords	WeeksRunning	Cost
CA005	2008/12/15	35	3	R 52.50
CO002	2008/11/20	16	2	R 16.00
CA006	2008/12/01	25	2	R 25.00
ST002	2008/11/26	12	2	R 12.00
PR001	2008/12/10	15	2	R 15.00
SP003	2008/12/10	18	2	R 18.00
CA001	2008/12/10	15	3	R 22.50
CO001	2008/11/23	12	2	R 12.00
PH001	2008/12/10	30	1	R 15.00
ST001	2008/12/11	35	1	R 17.50

:

(6)

1.8 Complete the SQL code in the **CountCellphones** button that will determine and display the number of clients with cellphone numbers. A cellphone number starts with 0 as the first digit. Example of output:

Number of clients with cellphone numbers
5

(4)

- Enter your examination number as a comment in the first line of the **NewspaperDBase\_U** unit containing the SQL statements.
- Save the unit named **NewspaperDBase\_U** and the project named **NewspaperDBase\_P** (File/Save All).
- Rename the folder **Question 1 Delphi** to **Quest1\_X**, where X should be replaced with your examination number.
- A printout of the code for the **NewspaperDBase\_U** unit will be required.

[40]

**QUESTION 2: DELPHI – OBJECT-ORIENTED PROGRAMMING**

At the local newspaper in town people can place classified advertisements as long as the advert does not exceed 255 words. The new editor of the newspaper is trying to do his bit in helping to fight pollution to conserve the environment. He has created new rules about how the cost of adverts is calculated. The rules are as follows:

- Adverts cost 50c per word.
- Words are separated by spaces. You can assume that the words will be separated by only one space. Example of an advertisement to be placed:

```
17" LCD monitor. Includes 4 port USB hub. 1 year warranty.  
R3799. Paul. 0834567899
```

- Words of 4 characters or less count as half a word. The advertisement shown above contains 8 half words and 6 full words. Therefore a total of 10 words will be charged for.
- Discount: Companies rated by the SABS as being eco-friendly, low-pollution companies pay half price. These advertisements contain the code word '**LowPolluter**'.

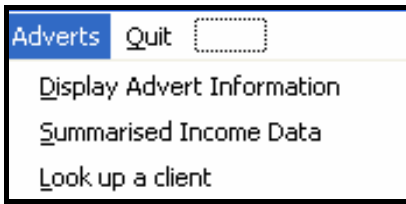
You have been asked to design a program that will make the process of working out the cost of an advert easier.

A text file named **Advertisements.txt** is supplied. The file contains the advertisements that must be placed. Each line within the text file consists of, firstly, the advertisement followed by the #-character and then the name of the client placing the advertisement. Example of the contents of the text file:

```
17" LCD monitor. Includes 4 port USB hub. 1 year warranty. R3799.  
Paul. 0834567899#P Weller  
LOST. Maltese poodle answers to cuddles. Yvette. 0821233211#Y  
WoefLieber  
Generator R12,499 - for when ESKOM lets you down. Price includes  
installation. 012 987 6543. ConsGoods#PowerBakup Inc  
iPod Video goggles. Watch your iPod movies on a virtual 80" screen.  
4 hour battery life. R2550.99 011 567 2344 HiTech Goodies  
Scrapbooking paper. Homemade. Large variety to choose from. All  
recycled. 012 609 8765 LowPolluter#The Scrapper  
:  
:
```

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 **testAdverts\_P.dpr** in the folder **Quest2\_X**.
- Go to File/Save As ... and save the unit as **testAdverts\_Uxxxx** (where xxxx represents the last FOUR digits of your examination number).
- Now go to File/Save Project As ... and save the project as **testAdverts\_Pxxxx** (where xxxx represents the last FOUR digits of your examination number).
- Change only the caption properties of the different options of the Menu component so that it corresponds with the figure below.



- 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 **Adverts\_UXXXX** and save this unit as **Adverts\_UXXXX** in your **Quest2\_X** folder. (XXXX should be replaced by the last FOUR digits of your examination number.) Write the following code as part of this class:

2.1.1 Define a class named **TAdvert**. This class must contain the following private fields:

**fAdvertText**  
**fClientName**

Ensure that you choose appropriate data types for these fields.

(4 ÷ 2) (2)

2.1.2 Write a parameterised constructor that will pass values for all the fields in the class. These parameters should be used to initialise the two fields of the class.

(4 ÷ 2) (2)

2.1.3 Write a method named **toString** that returns information on one advertisement in one string formatted as follows:

Name of the client, number of words, discount, total cost of the advertisement

Example of return string for the first advertisement in the text file **Advertisements**:

**Client: P Weller      Word Count: 10      No Discount      R5.00**  
(8 ÷ 2) (4)

- 2.1.4 Write a method named **countWords** which determines and returns the total number of billable words in the text of the advertisement. The codeword **LowPolluter** is not a billable word. Remember, words of 4 characters or less count as half a word. The total is always rounded up. (10 ÷ 2) (5)
- 2.1.5 Write a method named **isLowPolluter** that will return a boolean value indicating whether the client is a low polluter or not (in other words whether the advert text contains the word 'LowPolluter'). (6 ÷ 2) (3)
- 2.1.6 Write a method named **calculateCost** which calculates and returns the total cost of the advertisement. The client only pays for the text of the advertisement. Remember that low polluters get a discount. (4 ÷ 2) (2)
- 2.1.7 Write a **getClientName** method that will return the name of the client. (2 ÷ 2) (1)
- 2.2 Do the following in the **testAdverts\_UXXXX** unit (the main unit) in the given program:
- 2.2.1 Create an array named **arrAdverts** that keeps objects of **TAdvert** (there will not be more than 100 adverts). Write code in the **OnActivate EventHandler** of the form to read information from the text file **Advertisements** according to the following steps:
- (a) Test if the text file exists. If the file exists, open the text file and initialise a loop to read the data. Display a suitable message if the file does not exist and terminate the program.
  - (b) Read in a line of text from the text file.
  - (c) Separate the text into the advertisement text and client name.
  - (d) Use the name of the client and advertisement text to create a new **TAdvert** object and place the object into the array.
  - (e) Use a counter variable to keep track of how many objects were placed into the array. (16 ÷ 2) (8)

2.2.2 Write code to complete the options on the menu provided in the main unit of the given program. The methods in the **TAdvert** class should be called where applicable.

**Display Advert Information:** Display a suitable heading. Call the **toString** method to display the information on all the advertisements in the array. Example of the output:

Information on Advertisements			
Client: P Weller	Word Count: 10	No Discount	R 5.00
Client: Y WoefLieber	Word Count: 8	No Discount	R 4.00
Client: PowerBakup Inc	Word Count: 12	No Discount	R 6.00
Client: HiTech Goodies	Word Count: 15	No Discount	R 7.50
Client: The Scrapper	Word Count: 11	Discount	R 2.75
Client: PhotoPro	Word Count: 12	No Discount	R 6.00
Client: S Mthimunye	Word Count: 9	No Discount	R 4.50
Client: EconoTravel	Word Count: 14	No Discount	R 7.00
Client: SA Printers	Word Count: 12	Discount	R 3.00
Client: PhotoPro	Word Count: 14	No Discount	R 7.00
Client: PowerBakup Inc	Word Count: 11	No Discount	R 5.50
Client: P Weller	Word Count: 13	No Discount	R 6.50
Client: The Scrapper	Word Count: 12	Discount	R 3.00

(6 ÷ 2) (3)

**Summarised Income Data:** Display the total number of advertisements, the total number of half-price advertisements (low polluters) as well as the total number of full-price advertisements. Include the cost of each of these categories as part of the output. Use the relevant method(s) from the **Advert\_UXXXX** class. Example of the output:

Summary of income
13 advertisements in total worth R 67.75
3 half-priced advertisements worth R 8.75
10 full-priced advertisements worth R 59.00

(16 ÷ 2) (8)

**Look up a client:** Allow the user to enter the name of a client. Display all the advertisements placed by the client. A suitable message must be displayed if the name has not been found.

Example of output:  
User input: P Weller

Result of the search:			
Client: P Weller	Word Count: 10	No Discount	R 5.00
Client: P Weller	Word Count: 13	No Discount	R 6.50

(10 ÷ 2) (5)

- Enter your examination number as a comment in the first line of the main unit (**testAdverts\_UXXXX**) as well as the object unit (**Adverts\_UXXXX**).
- Save the files (File/Save All).
- Printouts of the code for the two units (**testAdverts\_UXXXX** as well as **Adverts\_UXXXX**) will be required.

**[43]**

### QUESTION 3: DELPHI PROGRAMMING

Mr Nkosi (chief editor) of the newspaper wants to run a competition called 'Charity Gold Rush'. He publishes a 4 x 4 grid of squares in the paper. Each block in the grid represents an unnamed charity. The public gets to buy a 'square' in the grid for R50 000 each.

He persuades the government to commit to randomly adding between 5% and 15% (both values inclusive) to the value of each square.

Five charities will be invited to benefit from the competition, namely:

- CHOC (represented by the letter 'C')
- CANSA (represented by the letter 'S')
- Aids Africa (represented by the letter 'A')
- Red Cross (represented by the letter 'R')
- The Association for Battered Women and Children (represented by the letter 'W')

He needs you to write a program that will do the following:

- The program will randomly allocate one of the five charities into each block in the grid starting from position [1,1]. It will also allocate a random value ranging from 5 to 15 that will represent the government's contribution to each block in the grid. For example:  
'A11' where the 'A' is the charity letter and the '11' is the random percentage that government will contribute.
- The program must then calculate how much money each charity gets (in other words R50 000 plus the government contribution for each block that the charity is found in).

The program must display the grid with the randomly allocated charity letters (C, S, A, R, W) and the total amount of money awarded to each charity.

Do the following:

- Create a folder named **Quest3\_X**. Replace X with your examination number.
- Open a new Delphi application.
- Save the unit (File/Save As) as **Charity\_UXXXX** and the project (File/Save Project As) as **Charity\_PXXXX** inside the **Quest3\_X** folder (XXXX should be replaced by the last FOUR digits of your examination number).
- Change the caption of the form to **Question 3 - X**. Replace X with your examination number.

You are required to do the following to design the program:

NOTE: Use appropriate components.

Do the following in the development of your program:

- 3.1 Define a two-dimensional array that will represent the competition grid. (2)
- 3.2 **Initialising the array:**  
Write a subprogram that will ensure that all the elements of the array are set to ' ' (a space). (3)
- 3.3 **Generating the random sales figures:**  
Write a subprogram to randomly assign a charity's representative letter AND a value between 5 and 15 (inclusive) to every square in the array. An example of what is stored in a square is 'A11' where the 'A' is the charity letter and the '11' is the random percentage that government will contribute. (10)
- 3.4 **Process and display the array:**  
The program needs to determine how much money each charity gets according to the contents of the array. Every time the charity's letter appears in the array it receives R50 000 PLUS the government's contribution as a percentage of R50 000.

The program needs to display the contents of the array in neatly tabulated columns on the screen. Below the array, the name of each charity, the total amount that they will receive, the amount that comes from the public and the amount that comes from the government must be displayed. Example of the output:



Grid with random values			
C13	R7	W13	A7
R14	A10	S10	S9
C14	C10	R14	A6
A13	C15	A12	A12
Amounts received per charity			
Total	Public	Government	Charity
R 226,000.00	R 200,000.00	R 26,000.00	CHOC
R 109,500.00	R 100,000.00	R 9,500.00	Cansa
R 330,000.00	R 300,000.00	R 30,000.00	Aids Africa
R 167,500.00	R 150,000.00	R 17,500.00	Red Cross
R 56,500.00	R 50,000.00	R 6,500.00	Battered Women & Children

Note: The values displayed on the screen will be different every time the program is executed.

(22)

- Enter your examination number as a comment in the first line of the unit **Charity\_UXXXX**.
- Save the unit and the project (File/Save All).
- A printout of the code for the unit **Charity\_UXXXX** will be required.

[37]

**TOTAL SECTION A: 120**

**SECTION B**

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

**QUESTION 1: JAVA – PROGRAMMING AND DATABASE**

The advertising department of the local newspaper has a classified section where readers can place advertisements on a weekly basis. They decided to make use of a database to assist them in managing the placement of these advertisements. A program is being developed to make it easier for the staff at the newspaper to answer certain queries being made frequently. You are required to assist in developing this program.

The information on clients placing advertisements as well as the advertisements being placed have been captured in a database named **NewspaperDB.mdb**.

The database, **NewspaperDB.mdb**, which contains two tables named **ClientsTb** and **AdvertsTb** respectively, has been supplied to you in a folder named **Question 1 Java**.

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

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

The **ClientsTb** table stores data on the clients of the newspaper. The fields in the **ClientsTb** table are defined as follows:

<u>Field Name</u>	<u>Type</u>	<u>Size</u>	<u>Comment</u>
ClientID	Text	5	Unique code of the client
ClientName	Text	30	
TelNum	Text	15	Contact number of the client
Suburb	Text	30	Suburb where the client lives

The following table (see next page) is an example of the data contained in the table called **ClientsTb**:

	<b>ClientID</b>	<b>ClientName</b>	<b>TelNum</b>	<b>Suburb</b>
+	BE013	Best Cars	(011) 678 9883	Bedfordview
+	CA012	CarSelect	(011) 098 3445	Roodepoort
+	DB002	DogBox	082 123 3210	Midrand
+	DO007	DOE	(011) 285 9887	Bedfordview
+	EC009	EconoTravel	(011) 542 3438	Johannesburg
+	HI004	HiTech Goodies	(011)567 2341	Roodepoort
+	JF008	Job For You	074 326 7851	Soweto
+	PE014	Pedigree Cars	(012) 343 9886	Centurion
+	PH006	PhotoPro	072 786 9880	Centurion
+	PO003	PowerBackup Inc	(012)987 6543	Pretoria Noord
+	PP001	The Pet Place	083 456 7891	Centurion
+	PR010	Property Wizard	084 009 8743	Pretoria
+	SC005	Scappers Heaven	(012) 609 8765	Pretoria Noord
+	SP011	SportsPro	(012) 025 6331	Centurion

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

<b><u>Field Name</u></b>	<b><u>Type</u></b>	<b><u>Size</u></b>	<b><u>Comment</u></b>
AdvertID	Text	5	Unique code of advert
ClientID	Text	5	Code of the client
DatePlaced	Date/Time	ShortDate	
WeeksRunning	Number	Integer	
NumWords	Number	Integer	Related to the advert placed
Paid	Boolean		Yes if paid. No if not paid
Category	Text	20	Category where advert will be placed

The following table (see next page) is an example of the data contained in the **AdvertsTb** table:

AdvertID	ClientID	DatePlaced	WeeksRunning	NumWords	Paid	Category
AN001	PP001	2008/10/19	2	15	<input checked="" type="checkbox"/>	Animals
AN002	DB002	2008/10/28	3	27	<input type="checkbox"/>	Animals
CA001	BE013	2008/12/10	3	15	<input checked="" type="checkbox"/>	Cars
CA002	BE013	2008/10/10	3	24	<input type="checkbox"/>	Cars
CA003	CA012	2008/10/24	2	14	<input type="checkbox"/>	Cars
CA004	CA012	2008/11/12	2	15	<input type="checkbox"/>	Cars
CA005	PE014	2008/12/15	3	35	<input type="checkbox"/>	Cars
CA006	BE013	2008/12/01	2	25	<input checked="" type="checkbox"/>	Cars
CO001	HI004	2008/11/23	2	12	<input checked="" type="checkbox"/>	Computers
CO002	HI004	2008/11/20	2	16	<input checked="" type="checkbox"/>	Computers
CO003	HI004	2008/11/16	3	14	<input checked="" type="checkbox"/>	Computers
CO004	HI004	2008/10/12	2	10	<input type="checkbox"/>	Computers
JO001	DO007	2008/10/12	1	35	<input checked="" type="checkbox"/>	Jobs
JO002	DO007	2008/11/01	1	30	<input checked="" type="checkbox"/>	Jobs
PH001	PH006	2008/12/10	1	30	<input checked="" type="checkbox"/>	Photographic
PR001	PR010	2008/12/10	2	15	<input checked="" type="checkbox"/>	Properties
PR002	PR010	2008/10/28	1	30	<input checked="" type="checkbox"/>	Properties
PR003	PR010	2008/11/08	2	24	<input type="checkbox"/>	Properties
SP001	SP011	2008/10/10	2	18	<input type="checkbox"/>	Sports
SP002	SP011	2008/11/12	2	20	<input checked="" type="checkbox"/>	Sports
SP003	SP011	2008/12/10	2	18	<input checked="" type="checkbox"/>	Sports
ST001	SC005	2008/12/11	1	35	<input type="checkbox"/>	Stationery
ST002	SC005	2008/11/26	2	12	<input type="checkbox"/>	Stationery

You have also been supplied with an incomplete Java program with a test class named **testNewspaper.java** and an object class named **Newspaper.java** in the **Question 1 Java** folder. Open the incomplete program. 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 nine options.

Do the following:

- Add your examination number as a comment above the first line of the given program.
- Change the code in the **testNewspaper** class to display the menu options, from top to bottom, to match the screenshot (FIGURE 1.1) shown below.

```

MENU

A - Display all the clients
B - Display all the Advertisements
C - Advertisements per category per month
D - Short and long Advertisements
E - Update Advertisements
F - Advertisements not paid for
G - Calculate costs of Advertisements
H - Count Clients with cellphones
Q - QUIT

Your choice_
    
```

FIGURE 1.1

- Change the names of the methods in the two Java files as follows:

**method A to displayAllClientsQuery(),**  
**method B to displayAllAdvertsQuery(),**  
**method C to selectCatMonthQuery(),**  
**method D to selectShortLongAdverts(),**  
**method E to updateAdverts(),**  
**method F to selectNotPaidAdverts(),**  
**method G to calcAdvertCost(),**  
**method H to countCellPhones().**

The connectivity code has already been written as part of the given code in the file named **Newspaper.java**. When you run the program, you have to enter the exact path where the database **NewspaperDB.mdb** has been stored.

HINT: Copy the database into the root directory of the drive that you are working on. The path to enter should be short, for example

E:/NewspaperDB.mdb.

HINT: Instead of entering the path every time that you run the program, you can change the input to a constant string containing the exact location of the database, for example

String file name = 'E:/NewspaperDB.mdb';

NOTE: If you cannot establish connectivity with the database at all 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 Newspaper.java.**

Complete the programming code in the file named **Newspaper.java** by creating the necessary SQL statements in the methods named **displayAllClientsQuery()**, **displayAllAdvertsQuery()**, **selectCatMonthQuery()**, **selectShortLongAdverts()**, **updateAdverts()**, **selectNotPaidAdverts()**, **calcAdvertCost()** and **countCellPhones()** respectively as indicated in QUESTIONS 1.1 to 1.8.

- 1.1 Complete the code in the **displayAllClientsQuery** method by formulating an SQL statement to display all the fields in the **ClientsTb** table in alphabetical order according to the names of the clients. Example of output:

ClientID	ClientName	TelNum	Suburb
BE013	Best Cars	(011) 678 9883	Bedfordview
CA012	CarSelect	(011) 098 3445	Roodepoort
D0007	DOE	(011) 285 9887	Bedfordview
DB002	DogBox	082 123 3210	Midrand
EC009	EconoTravel	(011) 542 3438	Johannesburg

(3)

:

- 1.2 Complete the code in the **displayAllAdvertsQuery** method by formulating an SQL statement to display all the fields in the **AdvertsTb** table except the **ClientID** field. Also display the **ClientName** field from the **ClientsTb** table. NOTE: You will need to link the tables with an appropriate **where/join** clause to be able to do this. Example of the output:

AdvertID	Date Placed	Weeks Running	Num of words	Paid	Category	Client Name
AN001	2008-10-19	2	12	True	Animals	The Pet Place
AN002	2008-10-28	3	27	False	Animals	DogBox
C0002	2008-11-20	2	16	True	Computers	HiTech Goodies
C0004	2008-10-12	2	10	False	Computers	HiTech Goodies
C0001	2008-11-23	2	12	True	Computers	HiTech Goodies
C0003	2008-11-16	3	14	True	Computers	HiTech Goodies
ST002	2008-11-26	2	12	False	Stationary	Scappers Heaven
ST001	2008-12-11	1	35	False	Stationery	Scappers Heaven
PH001	2008-12-10	1	30	True	Photographic	PhotoPro

(3)

:

:

- 1.3 Complete the code in the **selectCatMonthQuery** method, by asking the user to enter the name of the category and the number of the month as inputs. Formulate an SQL statement to display the **AdvertID**, **DatePlaced** and **WeeksRunning** fields for the adverts in the category and month entered by the user. Example of the output where the user enters 'Computers' as the category and 11 as the number of the month:

AdvertID	Date Placed	Weeks Running
C0002	2008-11-20	2
C0001	2008-11-23	2
C0003	2008-11-16	3

(6)

- 1.4 Complete the code in the **selectShortLongAdverts** method by formulating an SQL statement to display the **AdvertID**, the number of words and the names of the clients who placed adverts with less than 15 and more than 30 words. Display suitable headings. NOTE: You will need to link the tables with an appropriate **where/join** clause to be able to do this. Example of the output:

AdvertID	Number of Words	Client Name
AN001	12	The Pet Place
C0004	10	HiTech Goodies
C0001	12	HiTech Goodies
C0003	14	HiTech Goodies
ST002	12	Scappers Heaven
ST001	35	Scappers Heaven
J0001	35	DOE
CA003	14	CarSelect
CA005	35	Pedigree Cars

(6)

- 1.5 Complete the code in the **updateAdverts** method by asking the user to enter the ID of the advert and the number of words to be added. Formulate an SQL statement to update the number of words for the advert and then display the IDs and the number of words of all the advertisements after it has been updated.

(6)

- 1.6 Complete the code in the **selectNotPaidAdverts** method by asking the user to enter a letter of the alphabet. Formulate an SQL statement to display the IDs of the advertisements that have not been paid for yet in the categories starting with the letter entered by the user. Also display the names of the categories and the clients as well as the **Paid** field. NOTE: You will need to link the tables with an appropriate **where/join** clause to be able to do this. Example of the output if the user enters the letter S as input:

Advertisements not paid for			
AdvertID	Category	Name of Client	Has Paid
ST002	Stationery	Scappers Heaven	False
SP001	Sports	SportsPro	False
ST001	Stationery	Scappers Heaven	False

(6)

1.7 Complete the code in the **calcAdvertCost** method. Formulate an SQL statement to list the **AdvertID**, **DatePlaced**, **NumWords**, **WeeksRunning** and **Cost** of all the advertisements placed after 16 November 2008. **Cost** is a calculated field. The cost of an advertisement must be calculated at 50c per word per week. Example of output:

AdvertID	Date Placed	No. of Words	Weeks Running	Cost
CA005	2008-12-15	35	3	R 52.50
CO002	2008-11-20	16	2	R 16.00
CA006	2008-12-01	25	2	R 25.00
ST002	2008-11-26	12	2	R 12.00
PR001	2008-12-10	15	2	R 15.00
SP003	2008-12-10	18	2	R 18.00
CA001	2008-12-10	15	3	R 22.50
CO001	2008-11-23	12	2	R 12.00

:

(6)

1.8 Complete the SQL code in the **countCellphones** method that will determine and display the number of clients with cellphone numbers. A cellphone number starts with 0 as the first digit. Example of output:

```
Number of clients with cellphone numbers: 5
```

(4)

- Enter your examination number as a comment in the first line of the file named **Newspaper.java** containing the SQL statements.
- Save the **testNewspaper.java** and the **Newspaper.java** files.
- Rename the folder **Question 1 Java** to **Quest1\_X**, where X should be replaced with your examination number.
- A printout for the code of the **Newspaper.java** file will be required.

[40]



**QUESTION 2: JAVA – OBJECT-ORIENTED PROGRAMMING**

At the local newspaper in town people can place classified advertisements as long as the advert does not exceed 255 words. The new editor of the newspaper is trying to do his bit in helping to fight pollution to conserve the environment. He has created new rules about how the cost of adverts is calculated. The rules are as follows:

- Adverts cost 50c per word.
- Words are separated by spaces. You can assume that the words will be separated by only one space. Example of an advertisement to be placed:

```
17" LCD monitor. Includes 4 port USB hub. 1 year warranty.  
R3799. Paul. 0834567899
```

- Words of 4 characters or less count as half a word. The advertisement shown above contains 8 half words and 6 full words. Therefore a total of 10 words will be charged for.
- Discount: Companies rated by the SABS as being eco-friendly, low-pollution companies pay half price. These advertisements contain the code word '**LowPolluter**'.

You have been asked to design a program that will make the process of working out the cost of an advert easier.

A text file named **Advertisements.txt** is supplied. The file contains the advertisements that must be placed. Each line within the text file consists of, firstly, the advertisement, followed by the #-character and then the name of the client placing the advertisement. Example of the contents of the text file:

```
17" LCD monitor. Includes 4 port USB hub. 1 year warranty. R3799.  
Paul. 0834567899#P Weller  
LOST. Maltese poodle answers to cuddles. Yvette. 0821233211#Y  
WoefLieber  
Generator R12,499 - for when ESKOM lets you down. Price includes  
installation. 012 987 6543. ConsGoods#PowerBakup Inc  
iPod Video goggles. Watch your iPod movies on a virtual 80" screen.  
4 hour battery life. R2550.99 011 567 2344#HiTech Goodies  
Scrapbooking paper. Homemade. Large variety to choose from. All  
recycled. 012 609 8765 LowPolluter#The Scrapper  
:  
:
```

Do the following:

- Rename the folder **Question 2 Java** as **Quest2\_X** (where X represents your examination number).
- Open the **testAdvert** file (class) in the folder **Quest2\_X**.
- Change the options on the menu so that it corresponds with the menu shown below.

```

Menu

A - Display Advert Information
B - Summarised Income Data
C - Look up a client
Q - QUIT

Your choice:_

```

- Add your examination number as a comment in the first line of the **testAdvert** class.

2.1 Create an object class named **AdvertXXXX.java** and save this file as **AdvertXXXX.java** in your **Quest2\_X** folder. (XXXX should be replaced by the last four digits of your examination number.) This class should include code to do the following:

2.1.1 The object must contain the following private fields:

**advertText**  
**clientName**

Ensure that you choose appropriate data types for these fields.

(4 ÷ 2) (2)

2.1.2 Write a parameterised constructor that will pass all the values for the fields in the class. These parameters should be used to initialise the fields of the class.

(4 ÷ 2) (2)

2.1.3 Write a method named **toString()** that returns information on one advertisement in one string formatted as follows:

Name of the client, number of words, discount, total cost of the advertisement

Example of return strings for the first advertisement in the text file **Advertisements**:

**Client: P Weller Word count: 10 No Discount R5.0**

(8 ÷ 2) (4)

- 2.1.4 Write a method named **countWords** which determines and returns the total number of billable words in the text of the advertisement. The codeword **LowPolluter** is not a billable word. Remember, words of 4 characters or less count as half a word. The final total is always rounded up. (10 ÷ 2) (5)
- 2.1.5 Write a method named **isLowPolluter** which returns a boolean value indicating whether the client is a low polluter or not (in other words whether the text of the advertisement contains the word 'LowPolluter'). (6 ÷ 2) (3)
- 2.1.6 Write a method named **calculateCost** which calculates and returns the total cost of the advertisement. The client only pays for the text of the advertisement. Remember that low polluters get a discount. (4 ÷ 2) (2)
- 2.1.7 Write a **getClientName** method that will return the name of the client. (2 ÷ 2) (1)

2.2 Do the following in the **testAdvert** class of the given program:

- 2.2.1 Create an array named **arrAdverts** that keeps objects of **AdvertXXXX** (there will not be more than 100 adverts). Write code in the testclass to read information from the text file **Advertisements** according to the following steps:
- (a) Test if the text file exists. If the file exists, open the text file and initialise a loop to read the data. Display a suitable message if the file does not exist and terminate the program.
  - (b) Read in a line of text from the text file.
  - (c) Separate the text into the advertisement text and client name.
  - (d) Use the name of the client and advertisement text to create a new **AdvertXXXX** object and place the object into the array.
  - (e) Use a counter variable to keep track of how many objects were placed into the array. (8)
- 2.2.2 Write code to complete the options on the menu provided in the test class of the given program as indicated (see next page). The methods from the **AdvertXXXX** class should be invoked where applicable.

**Display Advert Information:** Display a suitable heading. Call the **toString** method to display the information on all the advertisements in the array. Example of the output (see next page):

```

Information on Advertisements

Client: P Weller           Word Count: 10  No Discount   R5.0
Client: Y WoefLieber      Word Count: 8   No Discount   R4.0
Client: PowerBakup Inc    Word Count: 12  No Discount   R6.0
Client: HiTech Goodies    Word Count: 15  No Discount   R7.5
Client: The Scrapper      Word Count: 11  Discount     R2.75
Client: PhotoPro          Word Count: 12  No Discount   R6.0
Client: S Mthimunye       Word Count: 9   No Discount   R4.5
Client: EconoTravel       Word Count: 14  No Discount   R7.0
Client: SA Printers       Word Count: 12  Discount     R3.0
Client: PhotoPro          Word Count: 14  No Discount   R7.0
Client: PowerBakup Inc    Word Count: 11  No Discount   R5.5
Client: P Weller          Word Count: 13  No Discount   R6.5
Client: The Scrapper      Word Count: 12  Discount     R3.0
    
```

(6 ÷ 2) (3)

**Summarised Income Data:** Display the total number of advertisements, the total number of half-price advertisements (low polluters) as well as the total number of full-price advertisements. Include the cost of each of these categories as part of the output. Use the relevant method(s) from the **AdvertXXXX** class. Example of the output:

```

Summary of income
13 advertisements in total worth R 67.75
3 half-priced advertisements in total worth R 8.75
10 full-priced advertisements in total worth R 59.0
    
```

(16 ÷ 2) (8)

**Look up a client:** Allow the user to enter the name of a client. Display all the advertisements placed by the client. A suitable message must be displayed if the name has not been found.

Example of output:  
User input: P Weller

```

Result of the search:
Client: P Weller           Word Count: 10  No Discount   R5.0
Client: P Weller          Word Count: 13  No Discount   R6.5
    
```

(10 ÷ 2) (5)

- Enter your examination number as a comment in the first line of the **testAdvert.java** file as well as the object class **AdvertXXXX.java**.
- Save the files.
- Printouts of the code for the two files **testAdvert.java** as well as **AdvertXXXX.java** will be required.

[43]

### QUESTION 3: JAVA PROGRAMMING

Mr Nkosi (chief editor) of the newspaper wants to run a competition called 'Charity Gold Rush'. He publishes a 4 x 4 grid of squares in the paper. Each block in the grid represents an unnamed charity. The public gets to buy a 'square' in the grid for R50 000 each.

He persuades the government to commit to randomly adding between 5% and 15% (both values inclusive) to the value of each square.

Five charities will be invited to benefit from the competition, namely:

- CHOC (represented by the letter 'C')
- CANSA (represented by the letter 'S')
- Aids Africa (represented by the letter 'A')
- Red Cross (represented by the letter 'R')
- The Association for Battered Women and Children (represented by the letter 'W')

He needs you to write a program that will do the following:

- The program will randomly allocate one of the five charities into each block in the grid starting with position [0,0]. It will also allocate a random value ranging from 5 to 15 that will represent the government's contribution to each block in the grid. For example: 'A11' where the 'A' is the charity letter and the '11' is the random percentage that government will contribute.
- The program must then calculate how much money each charity gets (in other words R50 000 plus the government contribution for each block that the charity is found in).

The program must display the grid with the randomly allocated charity letters (C, S, A, R, W) and value, as well as the total amount of money awarded to each charity.

Do the following:

- Create a folder named **Quest3\_X**. Replace X with your examination number.
- Create and save the test class as **testCharityXXXX** in the **Quest3\_X** folder (XXXX should be replaced by the last FOUR digits of your examination number).
- Create and save the object class as **CharityXXXX** in the **Quest3\_X** folder (XXXX should be replaced by the last FOUR digits of your examination number).
- Enter your examination number as a comment in the first line of each of the two classes.
- There will be no options to choose from in this program.

Do the following in the development of your program:

3.1 Define a two-dimensional array to represent the competition grid. (2)

3.2 **Initialising the array:**  
Ensure that all the elements of the array are set to ' ' (a space). (3)

3.3 **Generating the random sales figures:**  
Randomly assign a charity's representative letter AND a value between 5 and 15 (inclusive) to every square in the array. An example of what is stored in a square is 'A11' where the 'A' is the charity letter and the '11' is the random percentage that government will contribute. (10)

3.4 **Process and display the array:**  
The program needs to determine how much money each charity gets according to the contents of the array. Every time the charity's letter appears in the array it receives R50 000 PLUS the government's contribution as a percentage of R50 000.

The program needs to display the contents of the array in neatly tabulated columns on the screen. Below the array, the name of each charity, the total amount that they will receive, the amount that comes from the public and the amount that comes from the government must be displayed. Example of the output:

```

Grid with random values
R6 W13 S6 W14
R9 A13 A12 A6
C6 W14 S10 W10
R6 AB S6 W11

Amounts received per charity

      Total      Public  Government  Charity
R 219500.00 R 200000.00 R 19500.00  Aids Africa
R 53000.00  R 50000.00  R 3000.00   CHOC
R 160500.00 R 150000.00 R 10500.00  Red Cross
R 161000.00 R 150000.00 R 11000.00  CANSA
R 281000.00 R 250000.00 R 31000.00  Battered Women & Children
    
```

Note: The values displayed on the screen will be different every time the program is executed. (22)

- Enter your examination number as a comment in the first line of the **testCharityXXXX** and the **CharityXXXX** classes.
- Save the two classes.
- Make printouts of the code of the **testCharityXXXX** class and the **CharityXXXX** class to hand in.

[37]

**TOTAL SECTION B: 120**

**GRAND TOTAL: 120**