higher education & training

Department:
Higher Education and Training
REPUBLIC OF SOUTH AFRICA

NATIONAL CERTIFICATE (VOCATIONAL)

COMPUTER PROGRAMMING
(Second Paper)
NQF LEVEL 4

(10041024)

14 March 2018 (X-Paper)
09:00–13:00

Candidates are allowed to use student files.

This question paper consists of 18 pages.

NCV

TIME: 4 HOURS
MARKS: 100

INSTRUCTIONS AND INFORMATION

1. Answer ALL the questions.
2. Read ALL the questions carefully.
3. Number the answers according to the numbering system used in this question paper.

INSTRUCTIONS TO STUDENTS AND LECTURERS

1. Candidates must be seated in the computer laboratory 30 minutes before the start of the examination to ensure that all computers have the appropriate programs.
2. These questions must be completed on a computer using MS Office 2003 or a later version and VB.NET 2008 or VB.NET 2010.
3. Each answer must be printed after completing the question. Ensure that your EXAMINATION NUMBER appears on each PRINTOUT. NO questions without a printed EXAMINATION NUMBER will be marked.
4. In the event of problems such as a power failure, a computer breakdown or a printer breakdown, the invigilator will make the necessary arrangements to continue with the examination process once the problem has been resolved. Candidates will not be penalised with time lost under these circumstances.
5. All work on the computer should be saved at regular intervals to prevent loss of work. NO additional time may be allowed for work lost due to lack of saving.
6. NO candidate may print his/her work for another candidate, or make a memory stick available to another candidate or access another candidate's work on the network. Any attempt to access any information from or transfer information to another candidate in whatever manner is a contravention of the examination rules and regulations and will be viewed in an extremely serious light.

Copyright reserved

7. ALL PRINTOUTS or screen prints done during the session must be handed in.

8. Make sure that your EXAMINATION NUMBER appears on ALL PRINTOUTS.

9. Candidates may NOT copy the source code from VB.NET to MS Word and then print it from the word processor.

10. Steps to print a VB.NET form or any other screen:

    10.1 Run the solution.
    10.2 Hold the ALT key down and press the print screen button (This will copy the active screen).
    10.3 Open MS Word and paste the form/screen in a new document.
    10.4 Add your examination number to the top right-hand side of your page.
    10.5 Print the page.

11. You are allowed to make use of the help files in VB.NET.

12. The following files will appear on your hard drive in the folders:

    12.1 QUESTION 1
    C:\CP L4 \TicketsSales\ (Microsoft Visual Studio Solution)

    12.2 QUESTION 2
    C:\ CP L4 \SYSTEMS-POLICIES (Microsoft Visual Studio Solution)

    12.3 QUESTION 3
    para.txt
    …\Images\

    12.4 QUESTION 4
    Apply.html
    …\Images\

13. Candidates may use the following or any other program of their choice to create web pages:

    Notepad ++, Netbeans, Blue voda, Dreamweaver

Copyright reserved

QUESTION 1

The following questions refer to the VB.NET solution called *TicketsSales (C:\CP L4 QUESTION 1\TicketSales)* that has been saved on the hard drive.

A partially designed application for Easy Ticket that will help them to sell tickets easily by entering a client's cellphone number, selecting the tickets category (soccer, movie and theater) and selecting the number of tickets requested.

The application will calculate the amount payable for the tickets and display the amount of tickets ranges, as follows:

- Soccer R40,00 each
- Movie R75,00 each
- Theater R100,00 each

Below is the IPO-chart for the problem.

INPUT	PROCESSING	OUTPUT
cellnumber category nr_tickets	AmountPayable = nr_tickets x tPrice + 14% VAT	AmountPayable Tickets_reserved

The following STEPS explain how to use the program to calculate the *Tickets Sale*:

STEP 1 RUN the program and enter client ticket as FIGURE 1.1

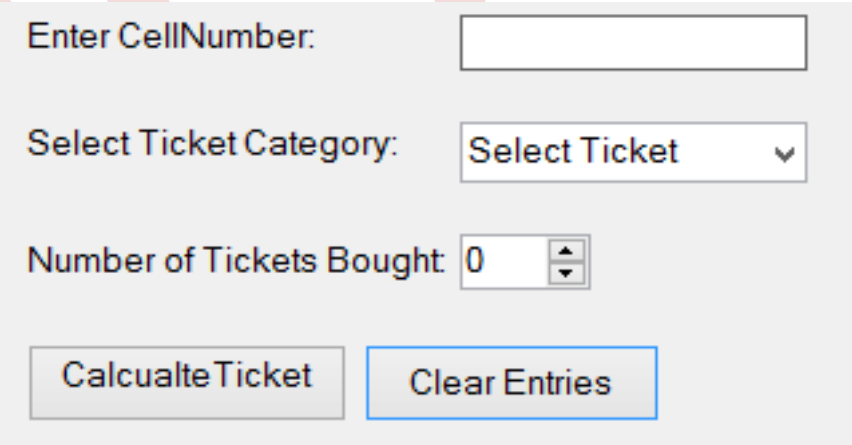


FIGURE 1.1

STEP 2 Enter the select ticket category and number of tickets as FIGURE 1.2

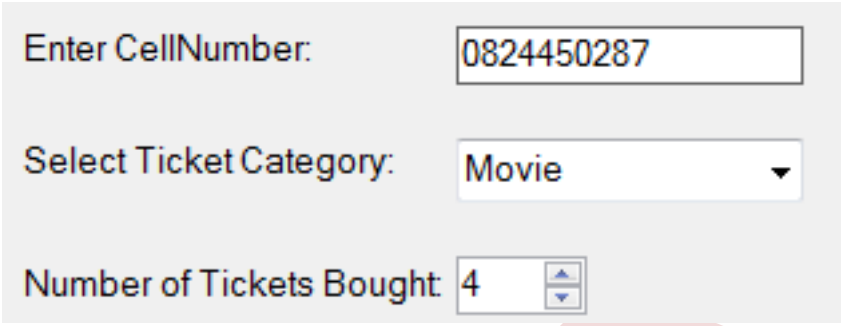


FIGURE 1.2

STEP 3 Click on the *Ticket Price* button to calculate and display the *Amount Payable* for the selected ticket category as FIGURE 1.3 below.

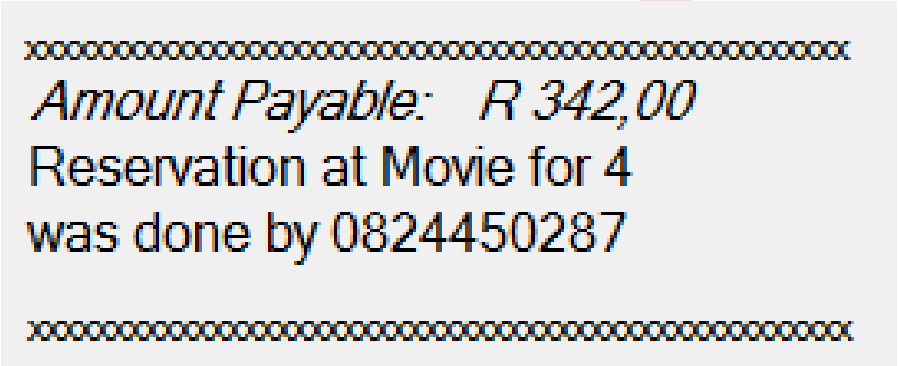


FIGURE 1.3

Copyright reserved

FIGURE 1.4 below is a screen print of what the user interface looks like. The completed interface has been saved on the hard disk.

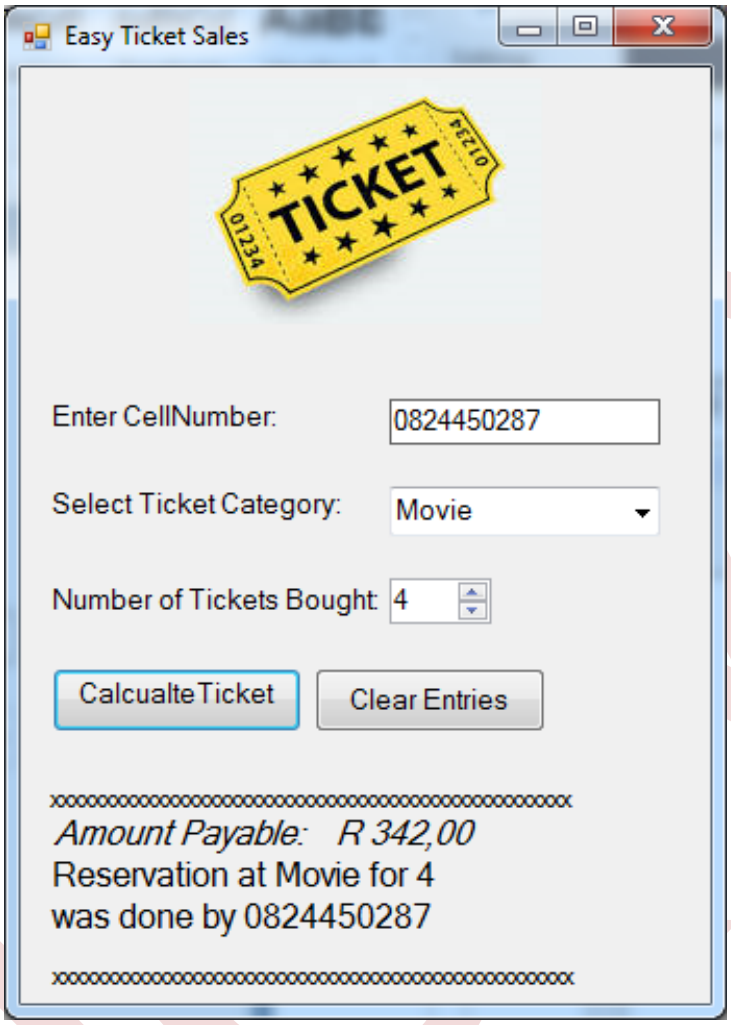


FIGURE 1.4

Below is the class diagram a class (*clsTicketSales*).
Study the class diagram FIGURE 1.5 and answer QUESTION 1.1

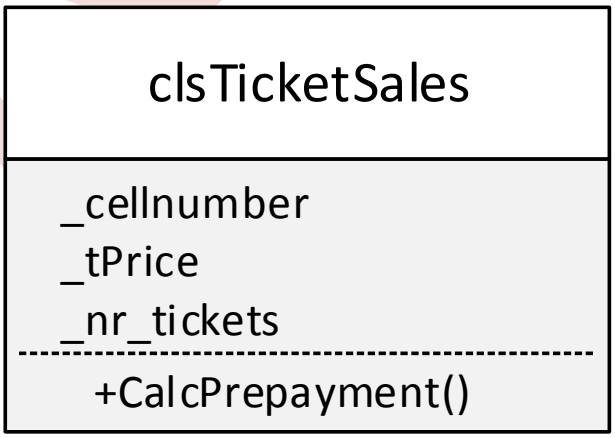


FIGURE 1.5

Copyright reserved

Open the solution *TicketSales* and follow the instructions to complete the program. The class clsTicketSales has been added to the project.

1.1 Add your EXAMINATION NUMBER to the program code as a comment. Add the following code to the *TicketSales* class named clsTicketSales:

1.1.1 Define the private variables for the clsTicketSales class. (3)

1.1.2 Write the code to create the following public properties for the TicketSales class:, each referring to set and retrieve the values of the private variables you defined for question 1.1.1

(a) cellNumber() (2)

(b) price() (1)

(c) nr_tickets()

Include error trapping code that will throw an exception from within the property procedure, in cases where the _nr_tickets field is to be set to zero.

An appropriate exception message should be displayed. (5)

1.1.3 Write the code to include a default constructor for the class. (2)

1.1.4 Write the code to include the CalcPrepayment() method for the *TicketSales* class. This method calculates and returns the amount due for the number of tickets bought including VAT at 14% (Refer to the IPO chart) (4)

Make a printout of the source code of class.

1.2 Add your EXAMINATION NUMBER as a comment to the program code of the frmTicketSales module. Add the following program code to the frmTicketSales:

Instantiate an *objTicke objectt* as a Public instance of the *clsTicketSales* class within the frmTicketSales form class. (2)

1.3 Add the code to the *Click Event* of the *btnCalculateSales* button:

1.3.1 (a) Determine which ticket category has been selected from the combobox.
Assign the values from the controls to the appropriate object properties. Make use of a Case – Select statement, to assign the correct price per ticket. (4)

(b) Call the CalcPrepayment*()* method of the *objTicket* class to calculate the amount payable of the ticket and display the answer in the appropriate labels. Refer to the example screenshot. Figure 1.4) (2)

Copyright reserved

Make a PRINTOUT of the source code.

Change the text in the title bar of frmTicketSales to your EXAMINATION NUMBER.

Run the program, enter the following test data and print frmTicketSales:

CellNumber	0794830054	0839950020	061782543
Category	*Movie*	Soccer	Theater
No_Ticket	*4*	5	2

[25]

QUESTION 2

You are requested to design a database query for Life Style to query the policy database to view policy members and remove member policy from the database.

A policy holder may have one or more policies as long as premiums are paid. A policy may be deleted at any time requested by the policy holder. There are five different policies to choose from, i.e. Life Cover, Car Insurance, House Cover, Education Cover, and Cellphone.

The VB. Net application will connect to an MS Access database that has already been created and populated with sample data. The database is saved on the hard drive in the application folder.

The database consists of two tables. Below is a graphical representation of the database tables and the relationship that exits between the tables:

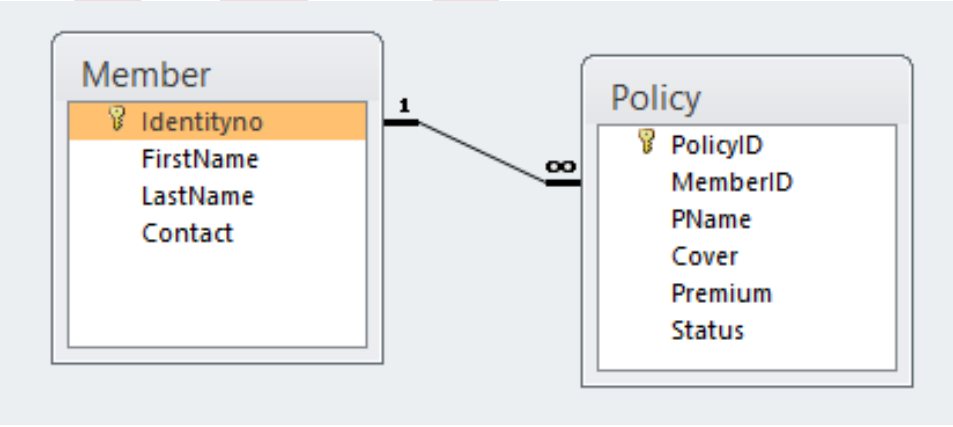


FIGURE 2.1

The structure of the two tables have been created as shown in FIGURE 2.2 and FIGURE 2.4:

TABLE 1: Member

Relationships | **Member** | Policy

Field Name	Data Type	Description
ID	Text	Identity number of the member (Primary key)
FirstName	Text	Name of policy member
LastName	Text	Surname of the member
Contact	Text	The cell phone number of the member

FIGURE 2.2

Below is a list of the records in the Table Member

Relationships | **Member** | Policy

ID	FirstName	LastName	Contact
6910315465798	SARALIE	STEYN	0872138532
7802312546488	JOHN	KHUMALO	0314425658
8112050245897	MARY	XULU	0727856810
8501284068486	JACK	BOTHA	0731042578
8809074518232	LEWIS	SMIT	0837502250
9107154758205	MATOME	LEDWABA	0718532921
9504078697897	JOY	JACKSON	0825412802
9605245464789	JACQUELINE	MARAISE	0686546712

FIGURE 2.3

TABLE 2: Policy

Member | **Policy**

Field Name	Data Type	
MemberID	Text	IdentityNr of the member
PolicyID	Text	Policy number of the member (Primary key)
PName	Text	Policy Name
Cover	Currency	Amount of the policy for a member
Premium	Currency	Amount to be paid by member
Status	Text	Active member or not

FIGURE 2.4

Copyright reserved

In the FIGURE 2.5 below is a list of some of the records in the TABLE Policy.

Relationships | Member | **Policy**

PolicyID	MemberID	PName	Cover	Premium	Status
1000012	8112050245897	Life Cover	R 350,000.00	R 450.00	active
1000022	8112050245897	Educational	R 75,000.00	R 150.00	active
1000023	8809074518232	Car Insurance	R 370,000.00	R 950.00	not active
100010	7802312546488	Educational	R 75,000.00	R 150.00	active
1000110	8112050245897	Car Insurance	R 300,000.00	R 750.00	active
100014	6910315465798	House Insurance	R 150,000.00	R 450.00	active
100015	6910315465798	Car Insurance	R 500,000.00	R 800.00	active
100016	7802312546488	Life Cover	R 250,000.00	R 300.00	active
100017	6910315465798	Life Cover	R 1,000,000.00	R 1,200.00	active
101010	7802312546488	Cellphone	R 3,500.00	R 45.00	active
1010106	8501284068486	Cellphone	R 4,500.00	R 99.00	not active

FIGURE 2.5

The following steps explain how to use the program to query the database and display the results for member's policy.

STEP 1 Run the program.
When the form is loaded, the program should display all the member policies in the *DataGridGiew*. See FIGURE 2.6.

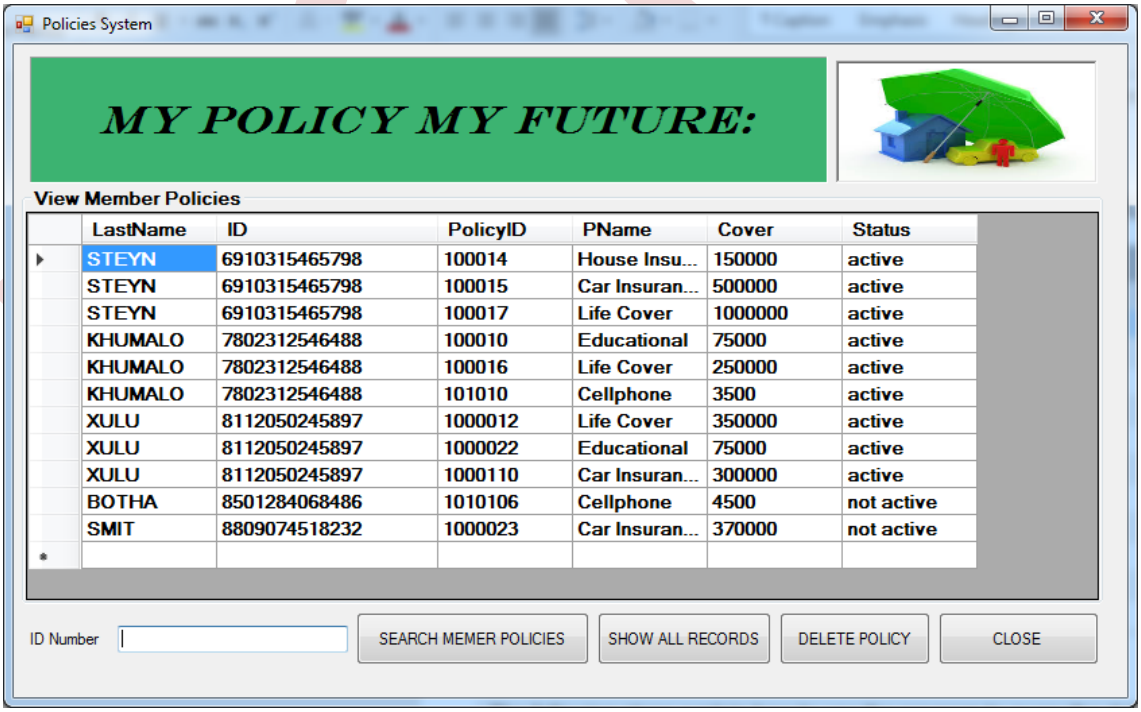


FIGURE 2.6

Copyright reserved

STEP 2 The user can search for all a members policies by typing in the ID number of the member and then clicking on the [Search Member Policies] button.

All the policies of the member should then be displayed as indicated in Figure 2.7. The user entered for example an ID Number of 8112050245897 and the [Search Member Policies] button is clicked all the corresponding policy records for that member is displayed.

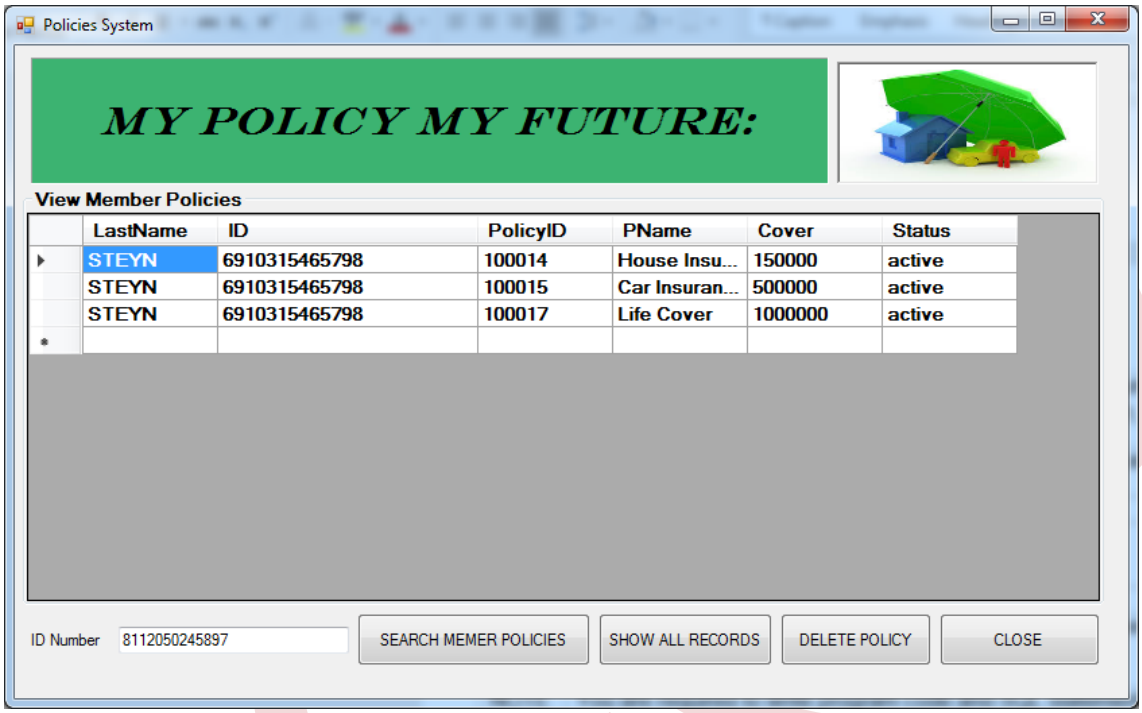
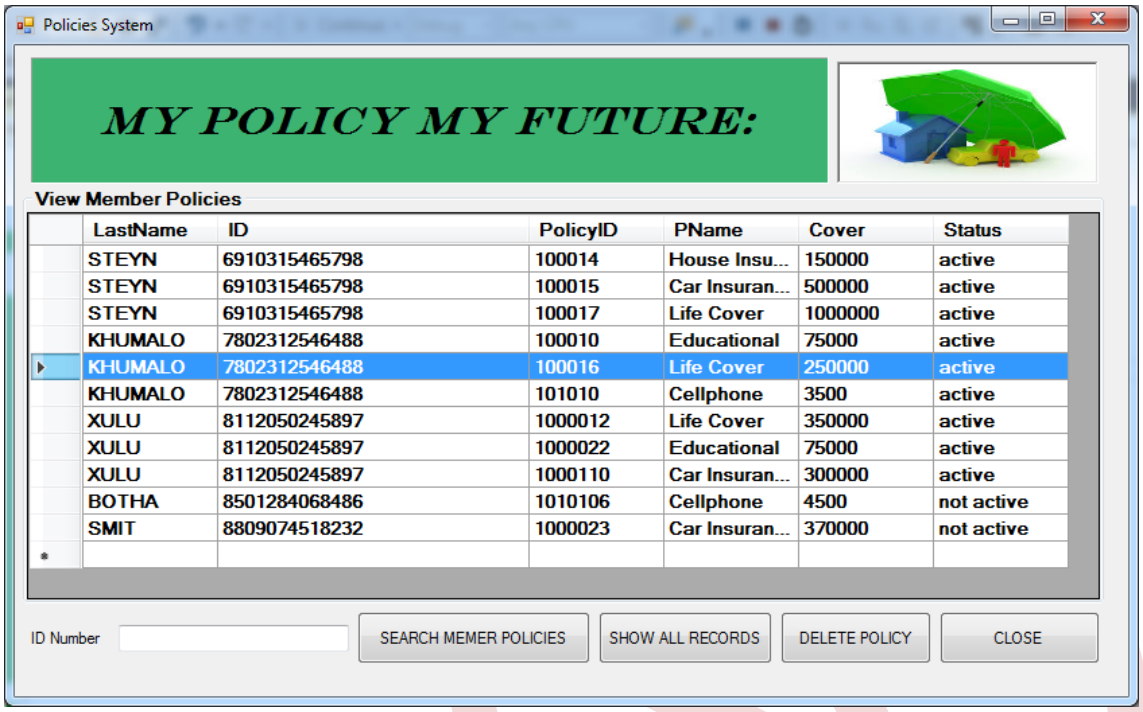


FIGURE 2.7

STEP 3 When a member policy is to be deleted.

When the user clicks on the [Delete Policy] button then the current members record's corresponding policy should be deleted. The user is given the option to confirm.

Copyright reserved



LastName	ID	PolicyID	PName	Cover	Status
STEYN	6910315465798	100014	House Insu...	150000	active
STEYN	6910315465798	100015	Car Insuran...	500000	active
STEYN	6910315465798	100017	Life Cover	1000000	active
KHUMALO	7802312546488	100010	Educational	75000	active
KHUMALO	7802312546488	100016	Life Cover	250000	active
KHUMALO	7802312546488	101010	Cellphone	3500	active
XULU	8112050245897	1000012	Life Cover	350000	active
XULU	8112050245897	1000022	Educational	75000	active
XULU	8112050245897	1000110	Car Insuran...	300000	active
BOTHA	8501284068486	1010106	Cellphone	4500	not active
SMIT	8809074518232	1000023	Car Insuran...	370000	not active

FIGURE 2.8

For example: In Figure 2.8 the current active record is Khumalo with the current policyID as 1000016. When the user clicks on the [Delete Policy] button the user must confirm whether the corresponding record in the Policy table must be deleted or not.

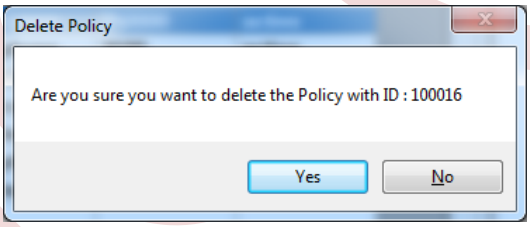


FIGURE 2.9

Copyright reserved

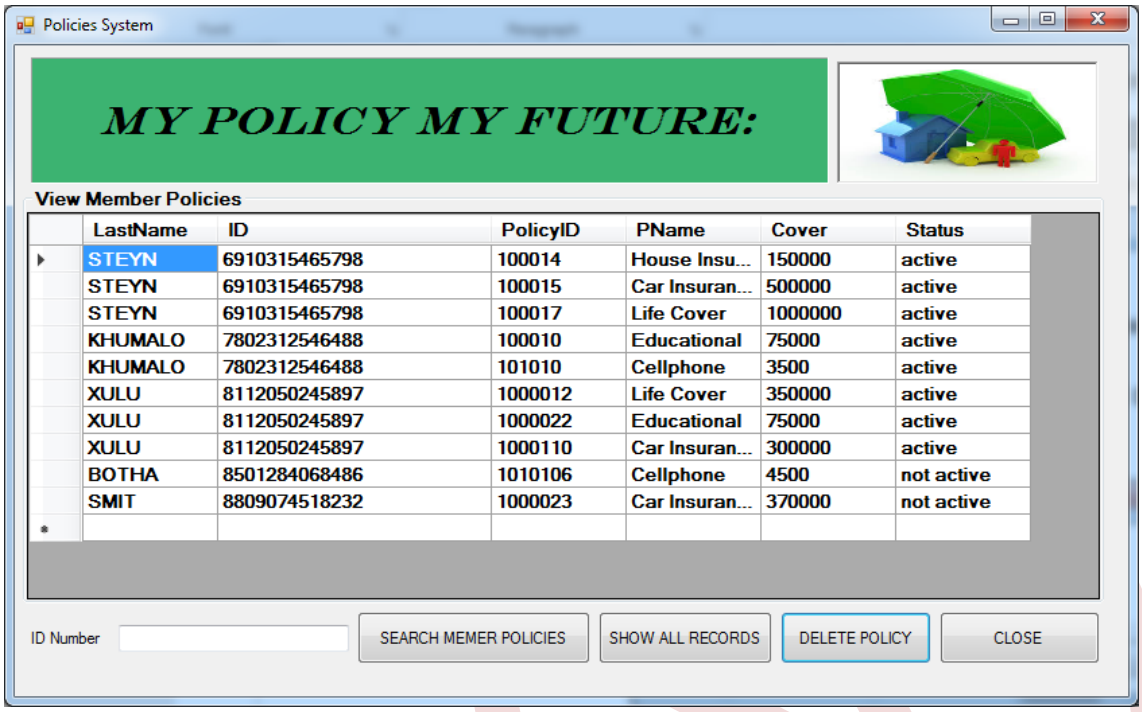


FIGURE 2.10

In Figure 2.10 the user selected Yes and the corresponding record is deleted and the table refreshed for display purposes.

The completed interface has been supplied to you. Open the solution and follow the instructions to complete the program.

NOTE You are required to write program code and SQL statements to complete the question. You may NOT use the DataSet Designer to complete the question.

2.1 Add the following program code to the GetPolicy() subprocedure:

2.1.1 Write the program code that will establish a connection to the database that would be needed to extract records from the tables. (5)

2.1.2 Write an SQL statement that will retrieve the all member's policy records from the database: member and policy tables.
(See STEP 1) The following fields should be shown: LastName, Identityno, PolicyID, PName, Cover, Status. TIP: Your SQL statement should include a Join (5)

2.1.3 When the frmPolicy is loaded the joined tables should be displayed in the DataGridView. (1)

Copyright reserved

2.2 Create a SearchPolicy() sub procedure and include a parameterized query that will be used to search for all the corresponding member policies. All policy and member details are listed where the ID number of the member is equal to the ID number field in the txtSearchNumber textbox (See FIGURE 2.7) (4)

When the btnSearchPolicy button is clicked SearchPolicy routine should be called and the Datagridview should be updated with the newest seleted fields. See STEP 2 (1)

2.3 Add the following program code to the DeletePolicy() subprocedure:

2.3.1 For the DeletePolicy() sub procedure, include a parameterized SQL statement that will delete the policy of the member selected. The policy corresponding to the current active record is deleted.

The user should confirm his/her decision using an appropriate messagebox (see example in STEP 3)

After a member has been deleted, a message box should show that the member has been deleted successfully.

The Datagridview should be updated after the delete to show all records still available (8)

2.3.2 Call the DeletePolicy() subprocedure in the *btndelete* click event. (1)

2.4 Run the program and delete the policy 100010 for member Khumalo.

Add your EXAMINATION NUMBER as a comment to the program code. Make a PRINTOUT of the program code.

[25]

QUESTION 3

You work for Be Financially Smart as a web developer. You have been tasked to create a web site for the business. The website consists of 2 pages: A home page (index.html) and an application page (apply.html). The purpose of the website is to help the youth with financial management. You have been given clear instructions on how to create each web page.

NOTE: You will design and create the home page (index.html) in QUESTION 3 and the application page (apply.html) in QUESTION 4

Below is a screen print of the home page FIGURE 3.1

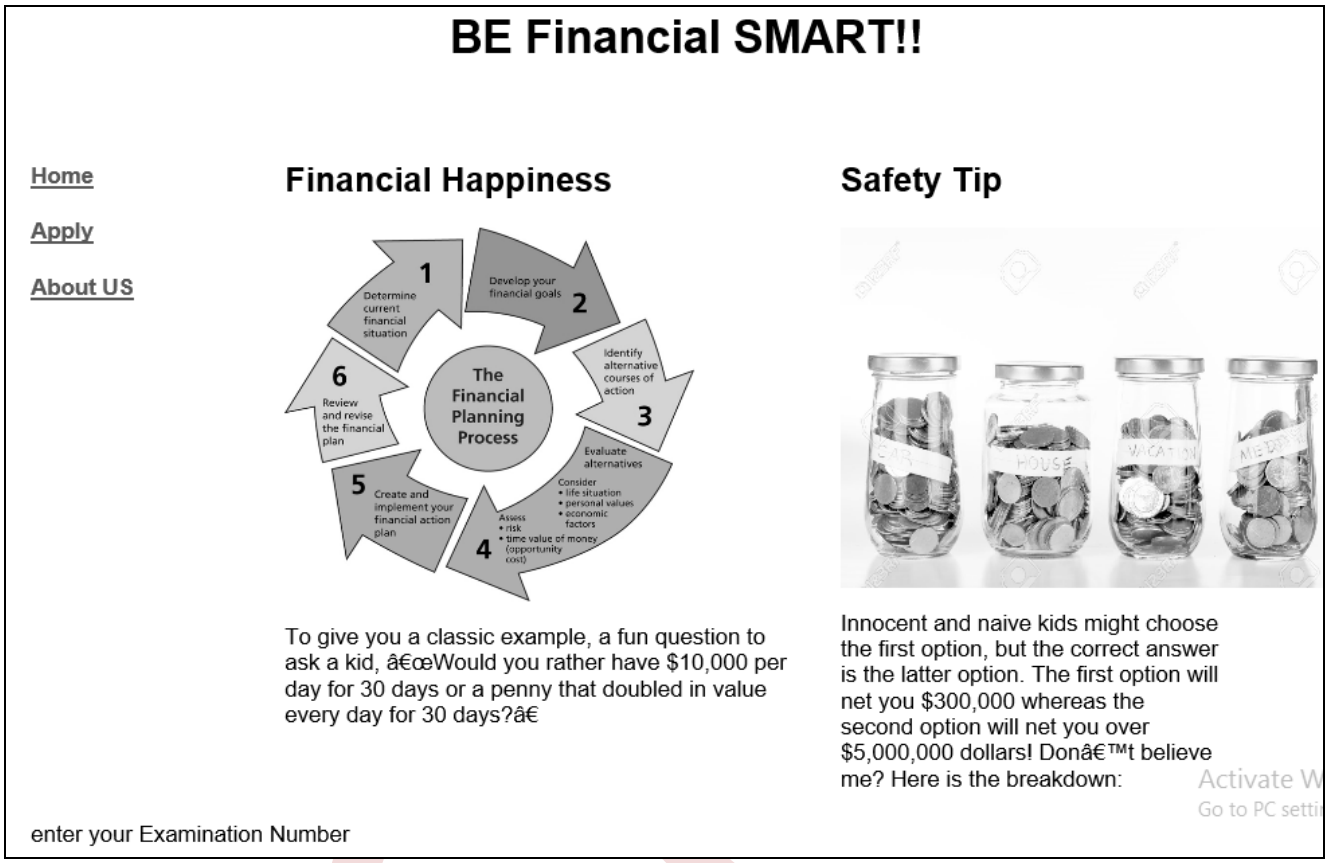


FIGURE 3.1

3.1 Create an external Cascading Style Sheet (CSS) and save it as *style.css.*

Add your EXAMINATION NUMBER as a comment to the CSS code.

3.1.1 Add a rule set for the header a height of 80 px, centre and padding of 0 15 px. (2)

3.1.2 Add a rule set for the content a width of 150 px float to left, padding of 5 px 15 px. (2)

3.1.3 Add a rule set for sidebar a width of 270 px, padding of 5 px 15 px and float to left. (2)

Make a PRINTOUT of the CSS code

Copyright reserved

3.2 Create an HTML file and save it as Index.html. Insert your EXAMINATION NUMBER as the title of the page.

3.2.1 Add a link to the external style sheet Style.css. (1)

3.2.2 Define the pagewrap division with six div elements. The first element is the header and it will contain the heading BFS in the body section (2)

3.2.3 Create the content division and insert the hyperlinks using a level 5 heading. (2)

3.2.4 Create the middle division and insert the heading using a level 1 heading, add the image save in the *folder\Question 3 \images.finance.jpg* with the height of 265 px and width of 290 px. (3)

Add the paragraph text saved in the *folder\Question 3\para.txt.* (1)

3.2.5 Create the sidebar division and insert the heading using a level 2 heading. Insert the image saved in the folder Question 3\images\saving.jpg with the width of 350 px and height of 255 px. (3)

3.2.6 Use the paragraph text saved in the *folder\Question 3\para.txt.* (1)

3.2.7 Create the footer division and insert your EXAMINATION NUMBER. (1)

Make a PRINTOUT of the source code.

Make a PRINTOUT of the web page from the web browser.

[20]

NCV

QUESTION 4

4.1 Below is a screen print of the Appllication page (*Apply.html*):



FIGURE 4.1

Insert the content for the Application page:

Insert your EXAMINATION NUMBER as the title to the web page.
Link the web page with the external style sheet called style2.css stored in

4.1.1 Create a borderless table to align the controls of the form. (2)

4.1.2 Create the web form as FIGURE 4.1 above. The textbox must accept a maximum length of 13 characters and contact must accept a maximum length of 10 characters (4)

4.1.3 Create a text area with 6 rows and 35 columns for suggestion and comments about the product. Add two buttons for Register and clear. (3)

4.1.4 Add the picture on the sidebar with a height of 255 px and a width of 350 px. (3)

Copyright reserved

4.1.5 Use JavaScript and create a user defined function to check whether the user has an e-mail in the e-mail textbox. (8)

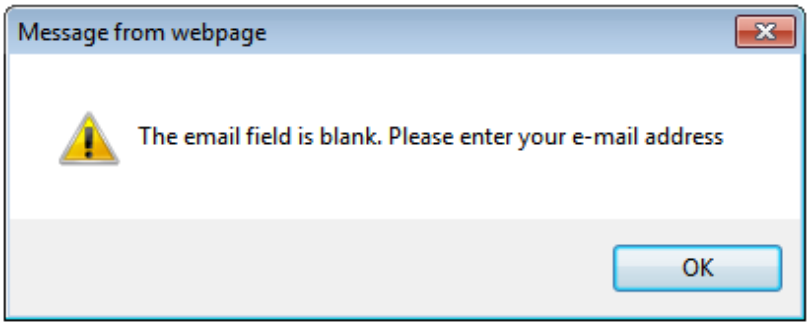


FIGURE 4.2

4.1.6 Display an appropriate m*essage* (See FIGURE 4.2 and set the focus into the textbox.) (2)

4.1.7 Write a code to check if the e-mail entered contains a "@" and ".". (5)

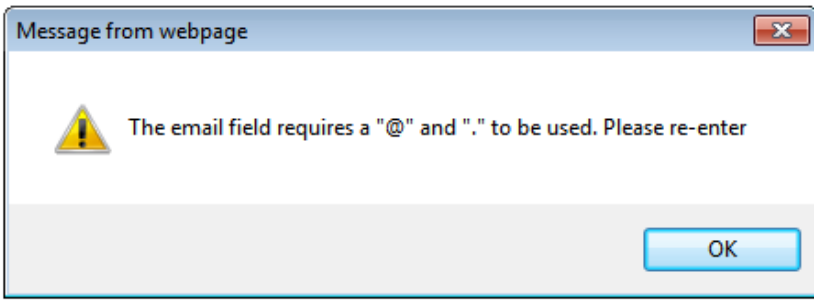


FIGURE 4.3

4.1.8 Display an appropriate message. (See FIGURE 4.3 and set the focus into the textbox and select the text.) (2)

4.1.9 When the user clicks the submit button display a message Thanking you for participating! (1)

The cancel button clears all the input boxes.

4.2 Make a PRINTOUT of the source code.

4.3 Print the web page from the web browser.

[30]

TOTAL: 100

Copyright reserved