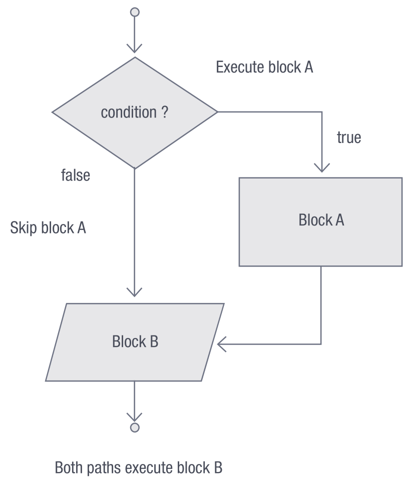
THE IF-THEN STATEMENT

IF and THEN are keywords in Delphi. To make a decision in your programming, you can use an IF-THEN statement in your code. The IF-THEN statement executes the statement/s following the THEN-keyword if the condition is TRUE and skips the execution of these statement/s when the condition is FALSE.

Take note

Remember that you cannot use IF and THEN as variable names. If you do, you will get an error 'Declaration expected but IF found'.

In the flow chart below, block A represents statement following the THEN-keyword. Pay careful attention to how block A is skipped if the condition is false.



SYNTAX OF IF-THEN STATEMENT

Below is an example of the Delphi syntax of an IF-THEN statement, if one statement follows the THEN-keyword:

```
If <condition> then
    <statement1>;
```

Note that the THEN-keyword it not followed by a semicolon because it is not the end of the statement. However, the <statement1> is followed by a semicolon, and ends the IF-THEN statement.

To help you to understand how to implement the IF-THEN statement using Delphi code, work through the following algorithm that determines if one number is a factor of another number.

Example 5.5 Determine points

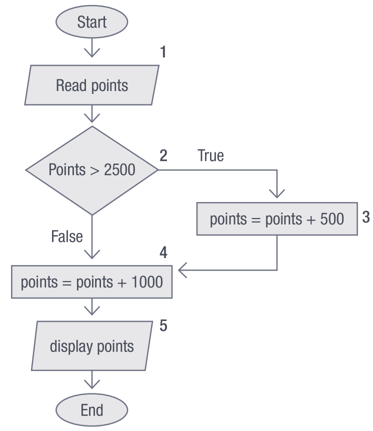
A customer is awarded points on a store card. The store decides to award 1000 bonus points to all its customers. Those customers who have points greater than 2500 before the bonus points are added, will be awarded an additional 500 points.

Create a flowchart that you can use to read a customer's current points on the store card, calculate the final points after the bonus points have been allocated, then displays the customer's final points.

IT-Practical-LB-Gr10 INK06.indb 94 2019/09/26 09:55

Example 5.5 Determine points *continued*

Read points
If points > 2500 then
    points = points + 500;
points = points + 1000;
Display points



If points equal 1200, trace through the owchart using the trace table below:

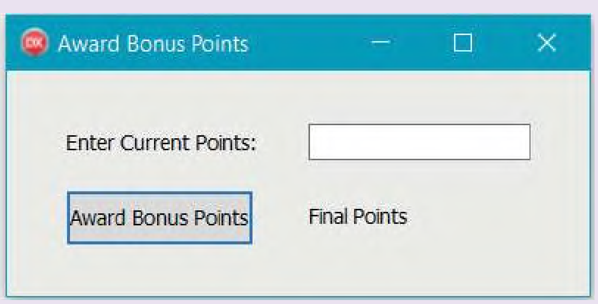
BOX NUMBER	POINTS	POINTS > 2500?	OUTPUT
1	1200		
2		False	
4	2200		
5			2200
Stop			

If points equal 3000, trace through the flowchart using the trace table below:

BOX NUMBER	POINTS	POINTS > 2500?	OUTPUT
1	3000		
2		True	
3	3500		
4	4500		
5			4500
Stop			

Now let's code the algorithm into Delphi.

1. Open the **AwardBonusPoints_p** project located in the 05 – Bonus Points folder.



IT-Practical-LB-Gr10 INK06.indb 95 2019/09/26 09:55

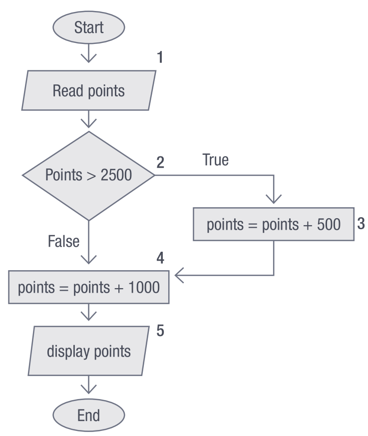
Example 5.5 Determine points *continued*

2. Create an OnClick event on [Award Bonus points] button to determine the final bonus points.

```
procedure TForm1.btnAwardPointsClick(Sender: TObject);
var
  iPoints : Integer;
begin
  iPoints := StrToInt(edtCurrentPoints.Text);
  if iPoints > 2500 then
    iPoints := iPoints + 500;  //end of IF statement
  iPoints := iPoints + 1000;
  lblFinalPoints.Caption := IntToStr(iPoints);
end;
```

Activity 5.3

Study the flowchart below and then answer the questions that follow:



5.3.1 If Box 4 is moved immediately below Box 1, will the algorithm work? Explain.

5.3.2 What will happen if Box 3 and Box 4 are interchanged?

5.3.3 Use trace tables to verify your answer to questions 1 and 2 above.

5.3.4 Amend the flowchart to display the original points, as well as the final points.

Activity 5.4

Write down Delphi conditional statements to do the following:

5.4.1 Set variable *sName* to 'John' if the value in variable *sSurname* equals to 'Karabo'.

5.4.2 Set variable *sEvenOrOdd* to 'Even' if the value in variable *iRemainder* equals 0.

5.4.3 Double the value in variable *iValue* if the value in variable *iInput* does not equal to 10.

5.4.4 Set variable *sGender* to 'Female' if the Boolean value in variable *bGender* is True.

5.4.5 Increase the value of variable *iTotal* by 10 if the value of variable *iTotal* is greater than or equal to 100.

Sometimes, if a condition is true, we need to execute more than one statement. Multiple statements are grouped together within the keywords Begin and End as shown below. It is seen as one group of statements to be executed in the THEN part.

Here is the syntax of an IF-THEN statement if more than one statement follows the THEN-keyword:

```
If <condition> then
begin
    <statement1>;
    <statement2>;
…
end;
```

Remember that the THEN and BEGIN keywords are not followed by a semicolon. However, the END statement marking the end of the IF-THEN statement is followed by a semicolon.

Example 5.6

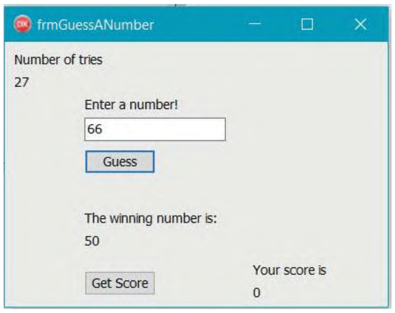
```
If Gender = 'F' then
begin
  iNumGirls := iNumGirls + 1;
  lblNumGirls.caption := IntToStr(iNumGirls);
end;
```

Activity 5.5

5.5.1 ABC stores have a promotion where a customer can win stars based on their spending. A customer will win one star for every R150 spent. If the customer spends more than R4000, they receive four extra stars. Customers cannot win part stars.
Open the **Reward_p** project located in the 05 – Stars Promo folder and complete the program by adding code to *btnCalculate* to determine how many stars a client must receive.
Test yourself: If a client enters 4150 your answer must be 31.

Watch out!
As a client cannot win part stars there must not be any decimal numbers. Refer to Chapter 4 to look at the difference between TRUNC and ROUND!

5.5.2 In this guessing game, a player guesses a number between 50 and 100.
The computer then displays a winning number. If the number is correct, a message with the words, 'Great Stuff' is displayed to congratulate the player AND 10 marks is added to the score.
If the number is 5 less than the winning number, 1 mark is added to the player's score and a message with the words, 'Keep going' is displayed.
A player may try as many times as they please.



IT-Practical-LB-Gr10 INK06.indb 97 2019/09/26 09:55

Activity 5.5 *continued*

a. Open the **Guess_p** project located in the 05 – Guess a Number folder.

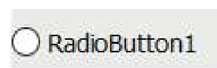
b. Complete the code for *btnGuess* as follows:
The winning number is generated randomly between 50 and 100. If the number guessed by the player is equal to the winning number, 10 is added to the score and a message is displayed in *lblMessage*.
If the number is 5 less than the winning number, 1 is added to the score and a message is displayed in *lblMessage*.

c. Complete the code for *btnGetScore* by displaying the score in *lblScore*.

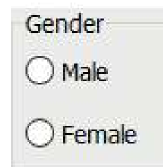
d. Run your program.
Hint: Use the variables provided.

THE RADIOBUTTON COMPONENT

A *RadioButton* component or option button that allows a user to choose only ONE button from a group of multiple buttons. To place a *RadioButton* on a form, you need to select the *TRadioButton* component from the *Standard Palette*.



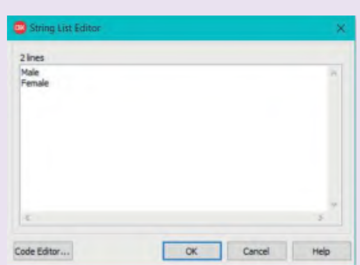
If you want to use radio buttons in groups of two or more – as shown in the figure alongside – the *TRadioGroup* component allows you to do this in a neat and dynamic way. When radio buttons are placed on a *TRadioGroup*, only ONE radio button can be selected at a time. If radio buttons are not placed in a *TRadioGroup*, then the radio buttons can be selected independently.



Example 5.7 Creating a group of RadioButtons

1. Select the *TRadioGroup* from the *Standard Palette* and place it on the form. The *TRadioGroup* serves as a container for the radio buttons.
2. Set the *Name* property of the *TRadioGroup* to *rgpGender*
3. Change the *TRadioGroup* caption to an appropriate name for the group of buttons, for example, 'Gender'.
4. Select the *TRadioGroup* component on the form. In the *Items* property in the *Object Inspector*, click on the ellipse (…).

5. In the *StringList Editor*, enter the names for the option buttons and click OK.

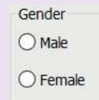


IT-Practical-LB-Gr10 INK06.indb 98 2019/09/26 09:55

Example 5.7 Creating a group of RadioButtons *continued*

6. Setting the *ItemIndex* property of the *TRadioGroup*:

   a. If the *ItemIndex* is set to -1, then none of the radio buttons will be selected at runtime.

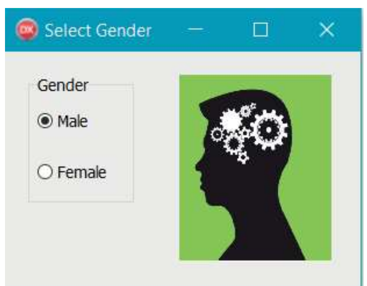


   b. If the *ItemIndex* is set to 0, then the first button is already selected at runtime.

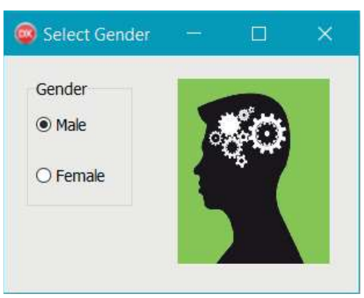
   c. If the *ItemIndex* is set to 1, then the second button is already selected at runtime.

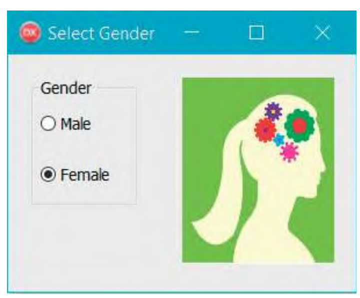
Guided Activity 5.1 Creating a group of RadioButtons

5.1.1 Open the **Gender_p** project from the 05 – Select Gender folder.



5.1.2 Create an *Onclick* event for the *RadioButton* group *rgpGender* so that when the Male option is selected, the boy image is displayed. When the female option is selected, then the girl image is displayed.





Guided Activity 5.1 Creating a group of RadioButtons *continued*

5.1.3 To create the *OnClick* event on the *rgpGender* group, double click on the *rgpGender* component. In the *rgpGender* event handler write the code (to load a picture into the image component during run time) for each option.

```
procedure TfrmGender.rgpGenderClick(Sender: TObject);
  begin
    if rgpGender.ItemIndex = 0 then
      imgGender.Picture.LoadFromFile('Male.bmp');

    if rgpGender.ItemIndex = 1 then
      imgGender.Picture.LoadFromFile('Female.bmp');
  end;
```

Image component ↓ | Property ↓ | Picture file name Add path name if file resides in different folder ↓

```
imgGender.Picture.LoadFromFile('File name')
```

5.1.4 Save and run your project.

Did you know

The *LoadFromFile* method allows you to load the contents of a file (the image) into the image component during run time.

Take note

- Previously, you set the ItemIndex property manually
- Now we are going to check the value of the ItemIndex property. The value of the ItemIndex property indicates which option is selected by the user.
- If the user selects the Male option, then ItemIndex will have the value 0 and if the Female option is selected, then the ItemIndex has a value 1
- We can test which option is selected by comparing ItemIndex against 0 or 1. Example: in the first IF-THEN statement the condition is: rgpGender.ItemIndex = 0. This tests if the current value of ItemIndex is equal to 0.
- If the Male option is selected, then the test rgpGender.ItemIndex = 0 evaluates to true and the Male image is loaded to the imgGender image component.
- Similarly, the female image is loaded when the Female option is selected

Activity 5.6

5.6.1 All library users pay a fee of R35.00 per month for the library services. To encourage reading amongst learners, they are offered the following discounts.

- Pre-schoolers get a discount of 15%
- Primary school learners get a discount of 13%
- High school learners and students get a discount of 11%.

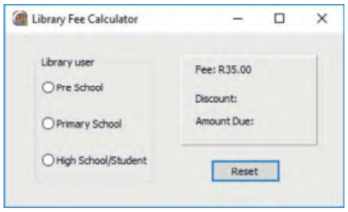
IT-Practical-LB-Gr10 INK06.indb 100 2019/09/26 09:55

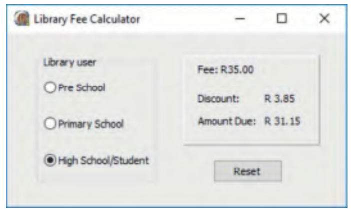
Activity 5.6 *continued*

a. Create a project, **Discount_p** to calculate the fees payable as follows:

- When the RadioButton in the RadioGroup is clicked, calculate the discount. Display the discount and amount due formatted as currency.
- Write code for a [Reset] button that will reset the RadioGroup so that no option is selected, and clear the labels so that nothing displays in the labels.

Use the interface shown below as a guide.

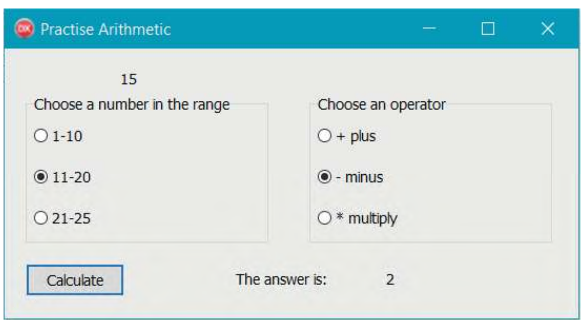




5.6.2 Open the **Arithmetic_p** project located in the 05 – Practise Arithmetic folder.

This program allows a user to practice their mental arithmetic. The RadioGroups allow a user to choose the number range and operator that they would like to practice.

Run the program and correct any errors you may encounter when the program executes.



Take note

Use the code `<labelname>.caption := '';` This places an empty string in the caption of the label – and displays nothing!

IT-Practical-LB-Gr10 INK06.indb 101 2019/09/26 09:55