**NATIONAL
SENIOR CERTIFICATE**

GRADE 12

SEPTEMBER 2023

**INFORMATION TECHNOLOGY P1
MARKING GUIDELINE**

MARKS: 150

This marking guideline consists of 20 pages.

NAME OF LEARNER:

TOTAL QUESTION 1	TOTAL QUESTION 2	TOTAL QUESTION 3	TOTAL QUESTION 4	TOTAL
/40	/35	/35	/40	/150

	QUESTION 1	MAX MARK	MARKS ACHIEVED
1.1	**BUTTON: [Q1.1 – Determine Cost]** Using an if / case statement ✓ Determine the license type from the radio group ✓ Display the license type ✓ as well as the cost of the license in the Cost label component ✓	**4**	

1.2.1	**BUTTON: [Q1.2.1 – Add the front wheel]**
	
	Create the dynamic shape: shpFrontWheel ✓
	Set parent of shpFrontWheel to the panel pnlQ1_2_Lights ✓
	
	Change the properties of shpFrontWheel:
	      Type → Circle ✓
	      Top → 25 ✓
	      Left → 35
	      Width → 50 ✓
	      Height → 50
	      Colour → White ✓
		6	

1.2.2	**BUTTON: [Q1.2.2 – Raise landing gear]** Change the colour of all three shapes to Red ✓	**1**	
1.2.3	**BUTTON: [Q1.2.3 – Lower landing gear]** Change the colour of all three shapes to Green ✓	**1**	

1.2.4	**BUTTON: [Q1.2.4 – Test landing gear]**
	
	Using either case or if statement, determine the state of the lights and produce the correct Message DLG
	
	If all 3 lights = Red ✓ → MessageDLG ✓ ('Catastrophic failure') ✓
	Else ✓
	  If all 3 lights = Green ✓ → MessageDLG('Safe to land') ✓
	Else ✓
	  MessageDLG('Caution, not safe to land') ✓
		8	

                            Please turn over

| 1.3 | **BUTTON: [Q1.3 – Top of Descent]**

Vertical Speed
  Get StartSpeed
  Get EndSpeed ✔
  Get Wind
  GetGlideSlope

AverageSpeed = (StartSpeed + EndSpeed) / 2 + Wind ✔✔
VerticalSpeed_NM_P_Min = (tan(GlideSlope * DEG_TO_RAD) *
                 AverageSpeed) / 60 ✔✔
VerticalSpeed := VerticalSpeed_NM_P_Min * NM_TO_FT ✔

Distance
  Get StartAltitude ✔
  Get EndAltitude

DeltaAltitude = StartAltitude – EndAltitude ✔
Distance = (DeltaAltitude * FT_TO_NM) / tan(GlideSlope *
          DEG_TO_RAD); ✔✔
DeltaSpeed = StartSpeed - EndSpeed; ✔
Distance = Distance + ceil(DeltaSpeed / 10) ✔
Distance = Distance + ceil(Wind / 10) ✔

Time
EstimatedTime = (Distance / AverageSpeed) * 60 ✔

Clear the rich edit
Add to rich edit:
 Vertical Speed: VerticalSpeed (fpm) - *formatted to 2 decimals* ✔✔
 Distance: Distance (nm) – *formatted to 2 decimals* ✔✔
 Estimated Time: EstimatedTime (min) – *formatted to 2 decimals* ✔✔ | **20** | |
| | **Question 1 Total** | **40** | |

QUESTION 2		MAX. MARK	MARKS ACHIEVED
2.1.1	**Button [Q2.1.1]** 'SELECT FirstName, Surname, Age, Email FROM tblPilots WHERE Age > 35 AND Age < 43 ORDER BY Age DESC' SELECT four correct fields ✓ FROM correct table ✓ WHERE Age in the range of 36 (incl) and 42 (incl) ✓ ORDER BY correct field DESC ✓	**4**	
2.1.2	**Button [Q2.1.2]** 'SELECT * FROM tblFlights WHERE Destination LIKE' + QuotedStr('%' + sLine + '%') SELECT * (all fields) ✓ FROM correct table ✓ WHERE Destination LIKE ✓ QuotedStr('%' + sLine + '%') ✓	**4**	
2.1.3	**Button [Q2.1.3]** 'SELECT count(*) AS [Flights in September] FROM tblFlights WHERE Month(DepartureDate) = 9' SELECT count(*) ✓AS [Flights in September] FROM tblFlights WHERE Month ✓ (DepartureDate) = 9 ✓	**3**	

2.1.4	**Button [Q2.1.4]**
	
	'SELECT Destination, Format(sum(PilotCostPerFlight), "Currency") AS [Pilot Cost], Format(sum(FlightCost), "Currency") AS [Flight Cost], Format(sum(PilotCostPerFlight) + sum(FlightCost), "Currency") AS [Total Cost] FROM tblPilots, tblFlights WHERE tblPilots.PilotID = tblFlights.PilotID GROUP BY Destination'
	
	SELECT Destination, ✓ ,
	  Format ✓ (sum ✓ (PilotCostPerFlight) ✓, "Currency"✓) AS [Pilot Cost] ✓,
	  Format(sum(FlightCost), "Currency") AS [Flight Cost], ✓
	  Format((sum(PilotCostPerFlight) ✓ + sum(FlightCost)), ✓
	  "Currency") AS [Total Cost]
	FROM both tables ✓ (tblPilots, tblFlights)
	WHERE link between tables ✓ (tblPilots.PilotID = tblFlights.PilotID)
	GROUP BY Destination ✓
		12	

2.1.5	**Button [Q2.1.5]** 'UPDATE tblPilots SET PilotCostPerFlight = PilotCostPerFlight * 1.07 WHERE LicenseType = "CPL"' UPDATE correct table ✓ SET PilotCostPerFlight = PilotCostPerFlight * 1.07 ✓ WHERE LicenseType = "CPL" ✓	**3**	

2.2.1	**Button [Q2.2.1]**
	if FieldByName('Age').AsInteger < 21 then ✓
	     delete ✓
	   else
	    next; ✓
		3	

2.2.2	**Button [Q2.2.2]**
	begin
	if Destination = sDestination then ✓
	  begin
	    if LicenseRequired = 'CPL' then
	     inc(iCPL)
	    else
	     if LicenseRequired = 'PPL' then ✓
	      inc(iPPL)
	      else
	      if LicenseRequired = 'MPL' then
	       inc(iMPL);
	    if CoPilotRequired = True then ✓
	     inc(iCoPilot)
	  end ✓ *NOTE: Co-Pilot must be inside begin and end, else it counts all the co-pilots and not just the destination co-pilots.*
	  Next ✓
	end
	Output ('CPL: ' + IntToStr(iCPL) + #13 +
	    'PPL: ' + IntToStr(iPPL) + #13 +
	    'MPL: ' + IntToStr(iMPL) + #13 + ✓
	    'Co-Pilots required: ' + IntToStr(iCoPilot));
		6	

	Question 2 Total	**35**	

QUESTION 3		MAX MARK	MARKS ACHIEVED

3.1.1	**Constructor Create**
	Correct heading and parameters ✓✓
	    fName := sName
	    fManufacturer := sManufacturer
	    fSpeed := rSpeed
	    fHeight := rHeight
	    fRange := rRange
	    fWeight := rWeight
	    fWingspan := rWingspan
	    fFirepower := iFirepower
	    fImageName := sImageName
	    fCountry := sCountry
	    fDescription := sDescription
	Correct assigning of all attributes ✓✓✓
		5	

3.1.2	**Mutator Method - setValues** Correct heading (procedure setValues) ✓ Correct conversion of each attribute ✓✓✓✓	**5**	

3.2.1	**OnChange event handler of cmbQ3_SelectAircraft**
	1. Extract the user's selection from the combo box ✓
	
	2. Test to see if the text file exists and assign the file. If the file does not exist, display a suitable message and exit
	    AssignFile(MyFile, 'Aircraft_List.csv'); ✓
	     Try ✓
	      Reset(MyFile); ✓
	     except
	      ShowMessage('File not found'); ✓
	      Exit; ✓
	     end; *Or alternative: if not (fileexists(textfile) = true) then*
	
	3. Loop through the text file until the user's selected aircraft has been found.
	    bFound := False; ✓
	    while (not eof(MyFile)) ✓ AND (bFound = False) ✓ do
	      if pos(UpperCase(sSearch),UpperCase(sOneline)) ✓ <> 0 ✓ then
	        bFound := True; ✓
	
	4. If the aircraft has been found, then:
	     Loop and extract the information from the text file
	     //Aircraft Name
	       iPos := pos(',',sOneLine); ✓
	       sName := copy(sOneLine,1,iPos-1); ✓
	       delete(sOneLine,1,iPos); ✓
	     //Other fields ✓✓
	
	     Instantiate (create) the object **objAircraft**.
	    objAircraft := TAircraft.Create ✓(sName, sManufacturer,
	        rSpeed, rHeight, rRange, rWeight, rWingspan,
	        iFirepower, sImageName, sCountry, sDescription); ✓✓
	
	5. Call the setValues method ✓
	6. Load the object data into the components ✓✓✓✓
		25	

	Question 3 Total	**35**	

QUESTION 4		MAX MARK	MARKS ACHIEVED
4.1	Extract the destination from the combo box ✓ Transfer/assign destination array to a2Booking ✓✓✓✓	**6**	

4.2	1. Extract the seat number ✓✓
	    Loop through the array to determine if the seat is booked ✓✓
	    Display error message if the seat has been booked ✓
	
	2. Book the seat in a2Booking ✓
	    Extract all the data ✓
	    Determine class ✓
	    Determine price ✓
	    If Business class * 1.95 ✓
	    Display heading Booking Information ✓ in bold ✓
	    Display booking ticket information neatly formatted:
	      'Name and Surname: ' + #13#9 + sNameSur + #13 + ✓
	      'Destination: ' + #13#9 + sDestination + #13 +
	      'Date and Time: ' + #13#9 + sDate + #13 +
	             #9 + sTime + ' flight' + #13 +
	      'Cabin: ' + #13#9 + sClass + #13 +
	      'Seat Number: ' + #13#9 + cCol ✓+ IntToStr(iRow + 1)✓+ #13 +
	      'Price: ' + #13#9 + FloatToStrF✓ (rPrice,ffCurrency✓,10,2);
		17	

4.3	1. Extract the destination ✓
	
	2. Loop through ar2Booking and increment the number of
	    passengers: Business Class
	        Loop through Rows 0 to 1 ✓
	         Loop through Cols 0 to 4 ✓
	          if ar2Booking[Row,Col] = 'B' then ✓
	            increment(Business Class Counter) ✓
	      and Economy Class
	        Loop through Rows 2 to 14 ⎤ ✓
	         Loop through Cols 0 to 4 ⎦
	          if ar2Booking[Row,Col] = 'B' then ⎤ ✓
	           increment(Economy Class Counter) ⎦
	
	3. Determine the cost for business class ✓✓ and economy class ✓✓
	
	4. Display booking ticket information neatly formatted:
	    'Passengers' + #13 +
	      #9 + 'Business Class: ' + IntToStr(iBusClass) + #13 + ✓
	      #9 + 'Economy Class: ' + IntToStr(iEcoClass) + #13 + ✓
	      #9 + 'Total: ' + IntToStr(iBusClass + iEcoClass) + #13#13 + ✓
	    'Cost' + #13 +
	      #9 + 'Business Class: ' + FloatToStrF(rBusPrice,ffCurrency,10,2) + #13 + ✓
	      #9 + 'Economy Class: ' + FloatToStrF(rEcoPrice,ffCurrency,10,2) + #13 + ✓
	      #9 + 'Total Cost: ' + FloatToStrF(rBusPrice + rEcoPrice,ffCurrency,10,2); ✓
		17	

	Question 4 Total	**40**	

SAMPLE SOLUTIONS

QUESTION 1

//////////   40 marks   //////////

//////////   Question 1.1 – 4 marks   //////////

```
procedure TfrmQuestion1.btnQ1_1_CostClick(Sender: TObject);
begin
  case rgpQ1_1_License.ItemIndex of
    0 : lblQ1_1_Cost.Caption := 'Microlight Pilot License = R37 000';
    1 : lblQ1_1_Cost.Caption := 'Private Pilot License = R110 451';
    2 : lblQ1_1_Cost.Caption := 'Commercial Pilot License = R761 379';
  end;
end;
```

//////////   Question 1.2.1 – 6 marks   //////////

```
procedure TfrmQuestion1.btnQ1_2_1Click(Sender: TObject);
begin
  shpFrontWheel := TShape.Create(frmQuestion1);
  shpFrontWheel.Parent := pnlQ1_2_Lights;

  with shpFrontWheel do
    begin
      Shape := stCircle;
      Top := 25;
      Left := 35;
      Height := 50;
      Width := 50;
      Brush.Color := clWhite;
    end;
end;
```

//////////   Question 1.2.2 – 1 mark   //////////

```
procedure TfrmQuestion1.btnQ1_2_2_UpClick(Sender: TObject);
begin
  btnQ1_2_1.Click; //Provided code, DO NOT DELETE
  shpFrontWheel.Brush.Color := clRed;
  shpLeftWheel.Brush.Color  := clRed;
  shpRightWheel.Brush.Color := clRed;
end;
```

//////////   Question 1.2.3 – 1 mark   //////////

```
procedure TfrmQuestion1.btnQ1_2_3_DownClick(Sender: TObject);
begin
  btnQ1_2_1.Click; //Provided code, DO NOT DELETE
  shpFrontWheel.Brush.Color := clGreen;
  shpLeftWheel.Brush.Color  := clGreen;
  shpRightWheel.Brush.Color := clGreen;
end;
```


////////// Question 1.2.4 – 8 marks //////////

```
procedure TfrmQuestion1.btnQ1_2_4_TestClick(Sender: TObject);
begin
  btnQ1_2_1.Click;   //Provided code, DO NOT DELETE
  RandomColours;   //Provided code, DO NOT DELETE

  if (shpFrontWheel.Brush.Color = clRed) AND
    (shpLeftWheel.Brush.Color = clRed) AND
    (shpRightWheel.Brush.Color = clRed) then
       MessageDLG('Catastrophic failure!',MTError,[MBOk],0)
  else
  if (shpFrontWheel.Brush.Color = clGreen) AND
    (shpLeftWheel.Brush.Color = clGreen) AND
    (shpRightWheel.Brush.Color = clGreen) then
       MessageDLG('Safe to land',MTInformation,[MBOk],0)
  else
    MessageDLG('Caution, not safe to land',MTInformation,[MBOk],0);
```

ALTERNATE SOLUTION
```
var
  iFront, iLeft, iRight, iTotal : Integer;

  iFront := 0;
  iLeft  := 0;
  iRight := 0;
  iTotal := 0;
  if shpFrontWheel.Brush.Color = clGreen then
    iFront := 1;
  if shpLeftWheel.Brush.Color = clGreen then
    iLeft := 1;
  if shpRightWheel.Brush.Color = clGreen then
    iRight := 1;

  iTotal := iFront + iLeft + iRight;
  case iTotal of
    0     : MessageDLG('Catastrophic failure!',MTError,[MBOk],0);
    1,2   : MessageDLG('Caution, not safe to land',MTInformation,[MBOk],0);
    3     : MessageDLG('Safe to land',MTInformation,[MBOk],0);
  end;
end;
```

////////// Question 1.3 – 20 marks //////////
procedure TfrmQuestion1.Q1_3_TopOfDescentClick(Sender: TObject);

```
const DEG_TO_RAD = 0.0174532925;
const NM_TO_FT = 6076.11549;
const FT_TO_NM = 1 / NM_TO_FT;

var
  rAverageSpeed, rStartSpeed, rEndSpeed, rVerticalSpeed, rWind  : Real;
  rGlideSlope, rVerticalSpeed_NM_P_Min                          : Real;
  rStartAltitude, rEndAltitude                                 : Real;
  rDistance, rDeltaAltitude, rDeltaSpeed                        : Real;
  rEstimatedTime                                               : Real;

begin
//Vertical Speed
  rStartSpeed := StrToFloat(edtQ1_3_StartSpeed.Text);
  rEndSpeed := StrToFloat(edtQ1_3_EndSpeed.Text);
  rWind := StrToFloat(edtQ1_3_Wind.Text);
  rGlideSlope := StrToFloat(edtQ1_3_GlideSlope.Text);

  rAverageSpeed := (rStartSpeed + rEndSpeed) / 2 + rWind;
  rVerticalSpeed_NM_P_Min := (tan(rGlideSlope * DEG_TO_RAD) * rAverageSpeed) / 60;
  rVerticalSpeed := rVerticalSpeed_NM_P_Min * NM_TO_FT;

//Distance
  rStartAltitude := StrToFloat(edtQ1_3_StartAltitude.Text);
  rEndAltitude := StrToFloat(edtQ1_3_EndAltitude.Text);

  rDeltaAltitude := rStartAltitude - rEndAltitude;
  rDistance := (rDeltaAltitude * FT_TO_NM) / tan(rGlideSlope * DEG_TO_RAD);
  rDeltaSpeed := rStartSpeed - rEndSpeed;
  rDistance := rDistance + ceil(rDeltaSpeed / 10);
  rDistance := rDistance + ceil(rWind / 10);

//Time
  rEstimatedTime := (rDistance / rAverageSpeed) * 60;

  redQ1_3.Clear;
  redQ1_3.Lines.Add('Vertical Speed: ' + FloatToStrF(rVerticalSpeed,ffFixed,10,2) + ' (fpm)' +
           #13 +
           'Distance: ' + FloatToStrF(rDistance, ffFixed, 10, 2) + ' (nm)' + #13 +
           'Estimated Time: ' + FloatToStrF(rEstimatedTime, ffFixed, 10, 2) + ' (min)');

end;
```

QUESTION 2

////////// 35 marks //////////

////////// Question 2.1.1 – 4 marks //////////

```
procedure TfrmQuestion2.btnQuestion2_1_1Click(Sender: TObject);
// Provided code - DO NOT DELETE OR ALTER //
var
  sSQL1: String;
begin
  /// Enter your code below ///
  sSQL1 := 'SELECT FirstName, Surname, Age, Email ' +
       'FROM tblPilots ' +
       'WHERE Age > 35 AND Age < 43 ' +
       'ORDER BY Age DESC';

  // Provided code - DO NOT DELETE OR ALTER //
  dbCONN.runSQL(sSQL1);
  if length(sSQL1) <> 0 then
    SetGridColumnWidths(dbgSQL);
end;
```

////////// Question 2.1.2 – 4 marks //////////

```
procedure TfrmQuestion2.btnQuestion2_1_2Click(Sender: TObject);
// Provided code - DO NOT DELETE OR ALTER //
var
  sSQL2: String;
  sLine : String;
begin
  /// Enter your code below ///
  sLine := InputBox('Destination','Enter the destination','');

  sSQL2 := 'SELECT * ' +
       'FROM tblFlights ' +
       'WHERE Destination LIKE ' + QuotedStr('%' + sLine + '%');

  // Provided code - DO NOT DELETE OR ALTER //
  dbCONN.runSQL(sSQL2);
  if length(sSQL2) <> 0 then
    SetGridColumnWidths(dbgSQL);
end;
```

////////// Question 2.1.3 - 3 Marks //////////

```
procedure TfrmQuestion2.btnQuestion2_1_3Click(Sender: TObject);
// Provided code - DO NOT DELETE OR ALTER //
var
  sSQL3 : String;
begin
  /// Enter your code below ///
  sSQL3 := 'SELECT count(*) AS [Flights in September] ' +
       'FROM tblFlights ' +
```

```
        'WHERE Month(DepartureDate) = 9';

  // Provided code - DO NOT DELETE OR ALTER //
  dbCONN.runSQL(sSQL3);
  if length(sSQL3) <> 0 then
    SetGridColumnWidths(dbgSQL);
end;




//////////    Question 2.1.4  –  12 marks     //////////
procedure TfrmQuestion2.btnQuestion2_1_4Click(Sender: TObject);
// Provided code - DO NOT DELETE OR ALTER //
var
  sSQL4: String;
begin
  /// Enter your code below ///
  sSQL4 := 'SELECT Destination, ' +
        'Format(sum(PilotCostPerFlight), "Currency") AS [Pilot Cost], ' +
        'Format(sum(FlightCost), "Currency") AS [Flight Cost], ' +
        'Format(sum(PilotCostPerFlight) + sum(FlightCost), "Currency") AS [Total Cost] ' +
        'FROM tblPilots, tblFlights ' +
        'WHERE tblPilots.PilotID = tblFlights.PilotID ' +
        'GROUP BY Destination';

  // Provided code - DO NOT DELETE OR ALTER //
  dbCONN.runSQL(sSQL4);
  if length(sSQL4) <> 0 then
    SetGridColumnWidths(dbgSQL);
end;




//////////    Question 2.1.5  –   3 marks      //////////
procedure TfrmQuestion2.btnQuestion2_1_5Click(Sender: TObject);
// Provided code - DO NOT DELETE OR ALTER //
var
  sSQL5: String;
begin
  /// Enter your code below ///
  sSQL5 := 'UPDATE tblPilots ' +
        'SET PilotCostPerFlight = PilotCostPerFlight * 1.07 ' +
        'WHERE LicenseType = "CPL"';

 // Provided code - DO NOT DELETE OR ALTER //
 dbCONN.executeSQL(sSQL5,dbgPilots,dbgFlights,dbgSQL);
 if length(sSQL5) <> 0 then
   SetGridColumnWidths(dbgSQL);
end;
```

```
//////////    Question 2.2.1  –  3 marks      //////////
procedure TfrmQuestion2.btnQuestion2_2_1Click(Sender: TObject);
begin
  // Provided code – DO NOT DELETE OR ALTER //
  redQ2_Output.Clear;

with tblPilot do
    begin
      Open;
      redQ2_Output.Lines.Add('Pilots before regulation change: ' + IntToStr(RecordCount));
      First;
      while not (eof) do
       begin
         /// Enter your code below ///
         if FieldByName('Age').AsInteger < 21 then
           Delete
         else
           Next;
        end;
      redQ2_Output.Lines.Add('Pilots after regulation change: ' + IntToStr(RecordCount));
    end;
end;


//////////    Question 2.2.2  -   6 Marks      //////////
procedure TfrmQuestion2.btnQuestion2_2_2Click(Sender: TObject);

// Provided code – DO NOT DELETE OR ALTER //
var
  sDestination : string;
  iCPL, iPPL, iMPL, iCoPilot : Integer;

begin

// Provided code - DO NOT DELETE OR ALTER //
  redQ2_Output.Clear;
  iCPL := 0;
  iPPL := 0;
  iMPL := 0;
  iCoPilot := 0;

 with tblFlight do
    begin
      Open;
      sDestination := cmbQ2_2_2_Destination.Text;
      redQ2_Output.SelAttributes.Style := [fsBold];
      redQ2_Output.Lines.Add(sDestination);
      redQ2_Output.Lines.Add('-------------------------');
      First;
```

```
while not (eof) do
    begin
      /// Enter your code below ///
      if FieldByName('Destination').AsString = sDestination then
        begin
          if FieldByName('LicenseRequired').AsString = 'CPL' then
            inc(iCPL)
            else
            if FieldByName('LicenseRequired').AsString = 'PPL' then
              inc(iPPL)
              else
              if FieldByName('LicenseRequired').AsString = 'MPL' then
                inc(iMPL);
          if (FieldByName('CoPilotRequired').AsBoolean = True) then
            inc(iCoPilot);
        end;
      Next;
    end;
    redQ2_Output.Lines.Add('CPL: ' + IntToStr(iCPL) + #13 +
                'PPL: ' + IntToStr(iPPL) + #13 +
                'MPL: ' + IntToStr(iMPL) + #13 +
                'Co-Pilots required: ' + IntToStr(iCoPilot));
  end;
end;
```

QUESTION 3

//////////   35 marks   //////////

//////////   Question 3.1.1 – 5 marks   //////////

```
constructor TAircraft.Create(sName, sManufacturer   : String;
                rSpeed, rHeight, rRange, rWeight     : Real;
                rWingspan                            : Real;
                iFirepower                           : Integer;
                sImageName, sCountry                 : String;
                sDescription                         : WideString);
begin
   fName          := sName;
   fManufacturer := sManufacturer;
   fSpeed         := rSpeed;
   fHeight        := rHeight;
   fRange         := rRange;
   fWeight        := rWeight;
   fWingspan      := rWingspan;
   fFirepower     := iFirepower;
   fImageName     := sImageName;
   fCountry       := sCountry;
   fDescription   := sDescription;
end;
```

//////////   Question 3.1.2 – 5 marks   //////////

```
procedure TAircraft.setValues;
begin
   fSpeed      := fSpeed * 1.852;       //Knots converted to KM/h
   fHeight     := fHeight * 0.3048;     //Feet converted to Meters
   fRange      := fRange * 1.60934;     //Miles converted to KM
   fWeight     := fWeight * 0.45359;    //Pounds converted to KG
   fWingspan   := fWingspan * 0.3048    //Feet converted to Meters
end;
```

//////////   Question 3.2.1 – 20 marks   //////////

```
procedure TfrmQuestion3.cmbQ3_SelectAircraftChange(Sender: TObject);
var
  sName, sManufacturer, sImageName, sCountry : String;
  rSpeed, rHeight, rRange, rWeight           : Real;
  rWingspan                                  : Real;
  iFirepower                                 : Integer;
  sDescription                               : Widestring;
  MyFile                                     : Textfile;
  sOneLine                                   : String;
  sSearch                                    : String;
  iPos                                       : Integer;
  bFound                                     : Boolean;
```



```
begin
  sSearch := cmbQ3_SelectAircraft.Text;

  AssignFile(MyFile, 'Aircraft_List.csv');
  try
    Reset(MyFile);
  except
    ShowMessage('File not found');
    Exit;
  end;

  bFound := False;
  Readln(MyFile, sOneLine); //Skip the heading line in the textfile
  while (not eof(MyFile)) AND (bFound = False) do
    begin
      Readln(MyFile, sOneLine);
      if pos(UpperCase(sSearch),UpperCase(sOneline)) <> 0 then
        begin
        //Aircraft Name
          iPos := pos(',',sOneLine);
          sName := copy(sOneLine,1,iPos-1);
          delete(sOneLine,1,iPos);
        //Manufacturer Name
          iPos := pos(',',sOneLine);
          sManufacturer := copy(sOneLine,1,iPos-1);
          delete(sOneLine,1,iPos);
        //Speed
          iPos := pos(',',sOneLine);
          rSpeed := StrToFloat(copy(sOneLine,1,iPos-1));
          delete(sOneLine,1,iPos);
        //Range
          iPos := pos(',',sOneLine);
          rRange := StrToFloat(copy(sOneLine,1,iPos-1));
          delete(sOneLine,1,iPos);
        //Weight
          iPos := pos(',',sOneLine);
          rWeight := StrToFloat(copy(sOneLine,1,iPos-1));
          delete(sOneLine,1,iPos);
        //Height
          iPos := pos(',',sOneLine);
          rHeight := StrToFloat(copy(sOneLine,1,iPos-1));
          delete(sOneLine,1,iPos);
        //Wingspan
          iPos := pos(',',sOneLine);
          rWingspan := StrToFloat(copy(sOneLine,1,iPos-1));
          delete(sOneLine,1,iPos);
        //Firepower
          iPos := pos(',',sOneLine);
          iFirepower := StrToInt(copy(sOneLine,1,iPos-1));
          delete(sOneLine,1,iPos);
        //Image Name
          iPos := pos(',',sOneLine);
          sImageName := copy(sOneLine,1,iPos-1);
```

Please turn over

```
       delete(sOneLine,1,iPos);
     //Country
       iPos := pos(',',sOneLine);
       sCountry := copy(sOneLine,1,iPos-1);
       delete(sOneLine,1,iPos);
     //Description
       sDescription := sOneLine;
     objAircraft := TAircraft.Create(sName,sManufacturer,rSpeed,rHeight,rRange,
                       rWeight,rWingspan,iFirepower,sImageName,sCountry,sDescription);
       bFound := True;
   end;
 end;

 if bFound then
   begin
     objAircraft.SetValues;
     lblQ3_AircraftName.Caption := objAircraft.getName;
     lblQ3_Manufacturer.Caption := objAircraft.getManufacturer;
     imgQ3_CountryFlag.Picture.LoadFromFile('Images\Flags\' + objAircraft.getCountry);
     imgQ3_AircraftImage.Picture.LoadFromFile('Images\Aircraft\' +
           objAircraft.getImageName);
     lblQ3_AircraftDescription.Caption := objAircraft.getDescription;
     lblQ3_MaxSpeed.Caption := FloatToStrF(objAircraft.getSpeed,ffFixed,10,0);
     lblQ3_MaxHeight.Caption := FloatToStrF(objAircraft.getHeight,ffFixed,10,0);
     lblQ3_Range.Caption := FloatToStrF(objAircraft.getRange,ffFixed,10,0);
     lblQ3_MaxTakeoffWeight.Caption := FloatToStrF(objAircraft.getWeight,ffFixed,10,0);
     lblQ3_Wingspan.Caption := FloatToStrF(objAircraft.getWingspan,ffFixed,10,2);
     lblQ3_Firepower.Caption := IntToStr(objAircraft.getFirepower);
   end;
end;
```

QUESTION 4

////////// 40 marks //////////

////////// Question 4.1 – 6 marks //////////

```
procedure TfrmQuestion4.cmbQ4_1_DestinationChange(Sender: TObject);
var
  iRow, iCol : Integer;
begin
  case cmbQ4_1_Destination.ItemIndex of //Case or If Statement
    0 : ar2Booking := ar2Bloemfontein;
    1 : ar2Booking := ar2CapeTown;
    2 : ar2Booking := ar2Durban;
    3 : ar2Booking := ar2EastLondon;
    4 : ar2Booking := ar2Johannesburg;
  end;

//  //Alternate Solution
//  case cmbQ4_Destination.ItemIndex of  //Case or If Statement
//    0 : for iRow := 0 to 14 do
//         for iCol := 0 to 4 do
//           ar2Booking[iRow,iCol] := ar2Bloemfontein[iRow,iCol];
//    1 : for iRow := 0 to 14 do
//         for iCol := 0 to 4 do
//           ar2Booking[iRow,iCol] := ar2CapeTown[iRow,iCol];
//    2 : for iRow := 0 to 14 do
//         for iCol := 0 to 4 do
//           ar2Booking[iRow,iCol] := ar2Durban[iRow,iCol];
//    3 : for iRow := 0 to 14 do
//         for iCol := 0 to 4 do
//           ar2Booking[iRow,iCol] := ar2EastLondon[iRow,iCol];
//    4 : for iRow := 0 to 14 do
//         for iCol := 0 to 4 do
//           ar2Booking[iRow,iCol] := ar2Johannesburg[iRow,iCol];
//  end;

  //Provide Code - DO NOT DELETE
  PaintColour;
end;


////////// Question 4.2 – 17 marks //////////
procedure TfrmQuestion4.pnlQ4_2_BookClick(Sender: TObject);
var
  sDestination, sDate, sNameSur, sTime : String;
  sClass, sLine : String;
  rPrice : Real;
  iRow, iCol : Integer;
  cCol : Char;
begin
```

```
 redQ4_Output.Clear;
 iRow := sedQ4_RowNumber.Value;

case cmbQ4_ColNumber.ItemIndex of
   0 : begin
         iCol := 0;
         cCol := 'A';
       end;
   1 : begin
         iCol := 1;
         cCol := 'B';
       end;
   2 : begin
         iCol := 3;
         cCol := 'C';
       end;
   3 : begin
         iCol := 4;
         cCol := 'D';
       end;
  end;

 if ar2Booking[iRow-1,iCol] = 'B' then
   MessageDLG('Seat has already been booked',MTInformation,[MBOK],0)
 else
 begin
   ar2Booking[iRow-1,iCol] := 'B';
   sDestination := cmbQ4_1_Destination.Text;
   sDate := DateToStr(dtpQ4_Date.Date);
   sTime := cmbQ4_Time.Text;
   sNameSur := edtQ4_NameSur.Text;
   sClass := 'Economy Class';
   rPrice := arrPrice[cmbQ4_1_Destination.ItemIndex];
   case iRow of
     1,2 : begin
           rPrice := rPrice * 1.95;
           sClass := 'Business Class';
         end;
   end;
   sLine := 'Name and Surname: ' + #13#9 + sNameSur + #13 +
         'Destination: ' + #13#9 + sDestination + #13 +
         'Date and Time: ' + #13#9 + sDate + #13 +
            #9 + sTime  + ' flight' + #13 +
         'Cabin: ' + #13#9 + sClass + #13 +
         'Seat Number: ' + #13#9 + cCol + IntToStr(iRow + 1) + #13 +
         'Price: ' + #13#9 + FloatToStrF(rPrice,ffCurrency,10,2);
   redQ4_Output.SelAttributes.Style := [fsBold];
   redQ4_Output.Lines.Add('Booking Information' + #13);
   redQ4_Output.Lines.Add(sLine);
 end;
PaintColour;
end;
```

```
//////////    Question 4.3 - 17 marks     //////////
procedure TfrmQuestion4.pnlQ4_3_StatsClick(Sender: TObject);
var
  sLine : String;
  iRow, iCol : Integer;
  iBusClass, iEcoClass : Integer;
  rBusPrice, rEcoPrice : Real;
begin
  redQ4_Output.Clear;
  redQ4_Output.Paragraph.TabCount := 1;
  redQ4_Output.Paragraph.Tab[0] := 10;
  iBusClass := 0;
  iEcoClass := 0;
  rBusPrice := 0;
  rEcoPrice := 0;
  for iRow := 0 to 1 do
    for iCol := 0 to 4 do
      begin
        if ar2Booking[iRow,iCol] = 'B' then
          inc(iBusClass);
      end;

  for iRow := 2 to 14 do
    for iCol := 0 to 4 do
      begin
        if ar2Booking[iRow,iCol] = 'B' then
          inc(iEcoClass);
      end;
  rBusPrice := iBusClass * (arrPrice[cmbQ4_1_Destination.ItemIndex] * 1.95);
  rEcoPrice := iEcoClass * (arrPrice[cmbQ4_1_Destination.ItemIndex]);
  redQ4_Output.SelAttributes.Style := [fsBold];

  sLine := 'Passengers' + #13 +
        #9 + 'Business Class: ' + IntToStr(iBusClass) + #13 +
        #9 + 'Economy Class: ' + IntToStr(iEcoClass) + #13 +
        #9 + 'Total: ' + IntToStr(iBusClass + iEcoClass) + #13#13 +
        'Cost' + #13 +
        #9 + 'Business Class: ' + FloatToStrF(rBusPrice,ffCurrency,10,2) + #13 +
        #9 + 'Economy Class: ' + FloatToStrF(rEcoPrice,ffCurrency,10,2) + #13 +
        #9 + 'Total Cost: ' + FloatToStrF(rBusPrice + rEcoPrice,ffCurrency,10,2);

  redQ4_Output.Lines.Add('Statistics of Flight' + #13);
  redQ4_Output.Lines.Add(sLine);
end;
```

TOTAL: 150