Province of the
EASTERN CAPE
EDUCATION

NATIONAL SENIOR CERTIFICATE

GRADE 12

SEPTEMBER 2021

INFORMATION TECHNOLOGY P1 MARKING GUIDELINE

MARKS: 150

This marking guideline consists of 16 pages.

NAME OF LEARNER:				
TOTAL QUESTION 1:	TOTAL QUESTION 2:	TOTAL QUESTION 3:	TOTAL QUESTION 4:	TOTAL
/40	/40	/38	/32	/150

QUESTION 1: GENERAL PROGRAMMING SKILLS		MAX. MARKS	MARKS ACHIEVED
1.1	**Button [1.1]** Load image to image file ✓ Enable pnlBath ✓ Colour of pnlRain changed to aqua ✓	**3**	
1.2	**Button [1.2 Process]** Get rain input from cmbRain or use the itemindex ✓ Extract the amount of millimetres: Check rain input for '/' ✓ If no '/' in input ✓ then get the number and convert to real ✓ Else ✓ Convert the amount of millimetres correctly for 1/10 ✓and 1/5 ✓ *Alternative solution:* *Use case with cmbRain itemindex to obtain the mm correctly ✓* *0 : set mm to 0.1 ✓* *1 : set mm to 0.5 ✓* *All other indexes set to correct mm amounts ✓✓* *End of case ✓* Get inputs from spinedits ✓ Calculate width * height * rain input / 150 ✓ Display in pnlBaths ✓ rounded to 1 decimal place ✓	**11**	
1.3	**Button [1.3 Find rainfall type]** Check edtRain for integer input ✓ Display a message ✓ and exit ✓ if not an integer ✓ *Alternative solution:* *Use val ✓* *If icode <> 0 ✓then showmessage ✓and exit ✓* Get input from edtRain ✓ Use case or nested if statements ✓ 0 : lblraintype.Caption := 'No rain'; ✓ 1 .. 3 : lblraintype.Caption := 'Moderate rain'; ✓ 4 .. 7 : lblraintype.Caption := 'Heavy rain'; ✓ 8 .. 9 : lblraintype.Caption := 'Very heavy rain'; ✓ 10 .. 50 : lblraintype.Caption := 'Heavy shower'; ✓ else ✓ lblraintype.caption := 'Flood'; ✓	**13**	

1.4	**Button [1.4 Display Rainfall]** Loop 8 times ✓ Use a loop ✓ to generate two different ✓ random numbers ✓ in range from 5 to 10 ✓ If a random number = 5 ✓ Then add 1 to LstQuantity index 0 ✓ If a random number = 10 ✓ Then add 1 to LstQuantity index 1 ✓ Calculate total quantity using LstQuantity items converted to integer ✓ Display total in LstQuantity index 2 converted to string ✓ Add LstQuantity values to heights of the two shapes ✓ Correct shapes using correct items converted to integer ✓	**13**	
	TOTAL QUESTION 1	**40**	

QUESTION 2: DATABASE PROGRAMMING		MAX. MARKS	MARKS ACHIEVED
2.1.1	**Button: [2.1.1]** SQL: 'select * from Status order by Statusname DESC' **Concepts:** SELECT correct field ✓ FROM correct table ✓ ORDER BY correct field DESC ✓	**3**	
2.1.2	**Button: [2.1.2]** SQL: 'Select Birdname from Bird where Sightings < 200' **Concepts:** SELECT correct field ✓ FROM correct table ✓ WHERE Sightings <200 ✓	**3**	
2.1.3	**Button: [2.1.3]** SQL: 'Select Birdname, Lastsighted from Bird where Birdname like ' + quotedstr('%' + sline + '%') **Concepts:** SELECT two correct fields ✓ FROM correct table ✓ WHERE Birdname LIKE ✓ quotedstr('%' + sline + '%') ✓	**4**	
2.1.4	**Button: [2.1.4]** SQL: 'Delete from Bird where (StatusID = 6) OR (StatusID = 5)' **Concepts:** DELETE from Bird ✓ WHERE StatusID = 6 ✓ **OR** ✓ StatusID = 5 ✓	**4**	
2.1.5	**Button: [2.1.5]** SQL: 'select Statusname, Birdname from Bird, Status where Bird.StatusID = Status.StatusID' **Concepts:** SELECT two correct fields ✓ FROM two correct tables ✓ WHERE clause to join two tables ✓	**3**	
2.1.6	**Button: [2.1.6]** 'Select StatusID,format(avg(Sightings),"fixed",1) as AverageSightings from Bird where year(Lastsighted) = 2007 group by StatusID' SELECT StatusID, ✓ format(avg ✓ (Sightings),"fixed",2) ✓ AS AverageSightings ✓ FROM Bird ✓ WHERE year(Lastsighted) ✓ = 2007 ✓ GROUP by StatusID ✓	**8**	
	2.1 Subtotal: SQL	**25**	

2.2.1	**Button: [2.2.1]** tblbird.First; ✓ while not tblbird.eof do ✓ if pos('EAGLE',uppercase(tblbird['Birdname'])) ✓ <> 0 then ✓ reddisplay.Lines.Add(tblbird['Birdname']); ✓ tblbird.Next; ✓	**6**	

2.2.2	**Button: [2.2.2]**
	
	icount := 0; ✓
	tblbird.First;
	while not tblbird.eof do ✓
	begin
	if tblbird['StatusID'] = 3 then ✓
	begin
	tblbird.Edit; ✓
	tblbird['StatusID'] := 2; ✓
	tblbird.Post; ✓
	inc(icount) ✓
	end;
	tblbird.Next; ✓
	end;
	reddisplay.Lines.Add('Number of changes made = ' + inttostr(icount)); ✓
		9	

	2.2 Subtotal: Code constructs	**15**	
	TOTAL QUESTION 2	**40**	

QUESTION 3: OBJECT-ORIENTATED PROGRAMMING		MAX. MARKS	MARKS ACHIEVED
3.1.1	**Constructor Create:** Correct name ✓ with four string parameters ✓ Set attributes to correct parameter values ✓ Set fQuantity to 1 ✓	**4**	
3.1.2	**function getsightings: integer;** Correct method – integer function ✓ Return correct attribute fQuantity ✓	**2**	
3.1.3	**procedure increasequantity(iqty: integer);** Correct method – procedure ✓ one integer parameter ✓ Add parameter ✓ to fQuantity ✓	**4**	
3.1.4	**function sightinggap: string;** Correct method – string function ✓ Get year ✓ from todays year ✓ Get year from date attribute ✓ Subtract attribute year from current year ✓ as integers ✓ Return result with correct words appended ✓	**7**	
3.1.5	**function tostring: string;** correct string method ✓ result returned with string compiled with correct attributes ✓ use #13, ✓ correct wording. ✓ Use SightingGap function ✓	**5**	
	3.1 Subtotal: Object class	**22**	
3.2.1	**Button [Q3.2.2]** Get inputs from two edits ✓ Get input from combobox ✓ and radiogroupbox ✓ Get inputs from spinedits ✓ Instantiate the object Object name = ✓ tsighting.create ✓ with four string parameters ✓ In correct order ✓ Display empty line in richedit ✓ Use method of class ✓ and object name to display ✓	**11**	
3.2.2	**Button [Q3.2.1]** Use object name ✓ and method of the class ✓ to increase quantity using sedSightings ✓ Display in richedit using object name ✓ and correct method ✓	**5**	
	3.2 Subtotal: Form class	**16**	
	TOTAL QUESTION 3	**38**	

QUESTION 4: PROBLEM SOLVING		MAX. MARKS	MARKS ACHIEVED

4.1	icount := 0; isum := 0; ✓
	assignfile(myfile, 'water.txt'); ✓
	reset(myfile); ✓
	while not eof(myfile) do✓
	readln(myfile, soneline); ✓
	inc(icount); ✓
	add type ✓ to arrtypes ✓
	add quantity to arrqty ✓
	isum := isum + arrqty[icount]; ✓
	
	for k := 1 to icount - 1 do ✓
	for l := k + 1 to icount do ✓
	if arrqty[k] > arrqty[l] then ✓
	itemp := arrqty[k];
	arrqty[k] := arrqty[l]; ✓
	arrqty[l] := itemp;
	stemp := arrtypes[k];
	arrtypes[k] := arrtypes[l]; ✓
	arrtypes[l] := stemp;
	
	for k := 1 to icount do ✓
	reddisplay.Lines.Add(arrtypes[k] ✓+ #9 +
	inttostr(arrqty[k])); ✓
	
	reddisplay.Lines.Add('Total number of people = ' +
	inttostr(isum)) ✓
		19	

4.2	for k := 1 to icount do ✓ iperc := round ✓ (arrqty[k] ✓ /isum *100 ✓); arrperc[k] := iperc; ✓ for k := 1 to icount do ✓ sline := arrtypes[k] + #9; ✓ case arrperc[k] of ✓ (use end of case correctly) 1..5 : sline := sline + 'X'; ✓ 6..10 : sline := sline +#9 + 'X'; ✓ 11..20 : sline := sline +#9 + #9 + 'X'; ✓ 21..50 : sline := sline +#9 + #9 + #9 + 'X'; ✓ end; reddisplay.lines.add(sline); inside the loop ✓	**13**	
	TOTAL QUESTION 4	**32**	

SAMPLE SOLUTIONS

QUESTION 1

```
procedure TfrmQuestion1.btnQ1_1Click(Sender: TObject); // 3 marks
begin
imgrain.picture.LoadFromFile('rain.jpg');
pnlbath.enabled := true;
pnlrain.Color := claqua;
end;

procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject); //  11 marks
var
rrain : real;
srain : string;
begin
 srain := cmbrain.Text;
 if copy(srain,2,1) <> '/' then
   begin
     rrain := strtofloat(copy(srain,1, pos(' ',srain) - 1));
   end
   else
   if copy(srain,3,1) = '1' then
      rrain :=  0.1
      else
      rrain := 0.5;
```



```
pnlbaths.Caption := floattostrf(sedheight.value * sedwidth.value * rrain / 150, fffixed,8,1);

end;

procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);  // 13 marks
var irain, inum, icode : integer;
begin
val(edtrain.Text, inum, icode);
 if icode <> 0 then
   begin
     showmessage('Enter only an integer');
     exit;
   end;
   irain := strtoint(edtrain.Text);
   case irain of
     0 : lblraintype.Caption := 'No rain';
     1 .. 3 : lblraintype.Caption := 'Moderate rain';
     4 .. 7 : lblraintype.Caption := 'Heavy rain';
     8 .. 9 : lblraintype.Caption := 'Very heavy rain';
     10 .. 50 : lblraintype.Caption := 'Heavy shower';
   else
    lblraintype.caption := 'Flood';
   end;

end;
```



```
procedure TfrmQuestion1.btnQ1_4Click(Sender: TObject);  // 13 marks
var iran1, iran2, k, itotal: integer;
begin
for k := 1 to 8 do
   begin
   repeat
   iran1 := randomrange(5,11);
   iran2 := randomrange(5,11);
   until (iran1 <> iran2);
   if iran1 = 5 then
      lstQuantity.Items[0] := inttostr(strtoint(lstQuantity.Items[0])+1);
   if iran2 = 10 then
      lstQuantity.Items[1] := inttostr(strtoint(lstQuantity.Items[1])+1);
   end;
 itotal := strtoint(lstQuantity.Items[0]) + strtoint(lstQuantity.Items[1]);
 lstQuantity.Items[2] := inttostr(itotal);
 shp5mm.Height := shp5mm.Height + strtoint(lstQuantity.Items[0]);
 shp10mm.Height :=  shp10mm.Height+ strtoint(lstQuantity.Items[1]);
end;

//provided code do not delete//////////
procedure TfrmQuestion1.FormActivate(Sender: TObject);
begin
lstQuantity.Items[0] := inttostr(0);
lstQuantity.Items[1] := inttostr(0);
lstQuantity.Items[2] := inttostr(0);
end;
////////////////////////////////////////
end.
```

QUESTION 2

```
/=============================================================================
//  Question 2.1.1   3 marks
//=============================================================================
==
procedure TQuestion_2.btnQuestion2_1_1Click(Sender: TObject);
var
  sSQL1: String;
begin
  sSQL1 := 'select * from Status order by Statusname DESC';
  // Provided code - do not change
  dbCONN.runSQL(sSQL1);

end;

//=============================================================================
==
//  Question 2.1.2   3 marks
//=============================================================================
==
procedure TQuestion_2.btnQuestion2_1_2Click(Sender: TObject);
// Provided code - do not change/
var
   sSQL2: String;

begin
  // Provided code - do not change//////////////////////

  sSQL2 := 'Select Birdname from Bird where Sightings < 200';

  // Provided code - do not change
  dbCONN.runSQL(sSQL2);
end;

//=============================================================================
==
//  Question 2.1.3   4 marks
//=============================================================================
==
procedure TQuestion_2.btnQuestion2_1_3Click(Sender: TObject);
// Provided code - do not change
var
 sline : string;
  sSQL3: String;
begin
   sline := inputbox('Enter a bird name','','vulture');

   sSQL3 := 'Select Birdname, Lastsighted from Bird where Birdname like '  + quotedstr('%'
+ sline + '%');

  // Provided code - do not change
  dbCONN.runSQL(sSQL3);
end;
```

```
//=============================================================
==
//  Question 2.1.4   4 marks
//=============================================================
==
procedure TQuestion_2.btnQuestion2_1_4Click(Sender: TObject);
// Provided code - do not change
var
  sSQL4: String;
begin

  sSQL4 := 'Delete from Bird where (StatusID = 6) OR (StatusID = 5)';

  // Provided code - do not change
  dbCONN.executeSQL(sSQL4,dbgstatus,dbgbird,dbgqrybird);
end;

//=============================================================
==
//  Question 2.1.5   3 marks
//=============================================================
==
procedure TQuestion_2.btnQuestion2_1_5Click(Sender: TObject);
// Provided code - do not change
var
  sSQL5: String;
begin

  sSQL5 := 'select Statusname, Birdname from Bird, Status where Bird.StatusID =
Status.StatusID' ;

 // Provided code - do not change
 dbCONN.runSQL(sSQL5);
end;

//=============================================================
==
//  Question 2.1.6   8 marks
//=============================================================
==
procedure TQuestion_2.btnQuestion2_1_6Click(Sender: TObject);
// Provided code - do not change
var
  sSQL6: String;
begin
  sSQL6 := 'Select StatusID,format(avg(Sightings),"fixed",1) as AverageSightings from Bird
where year(Lastsighted) = 2007 group by StatusID';
 // Provided code - do not change
 dbCONN.runSQL(sSQL6);

end;
```

```
//=============================================================================
==
//  Question 2.2.1   6 marks
//=============================================================================
==
procedure TQuestion_2.btnQuestion2_2_1Click(Sender: TObject);
begin
reddisplay.Clear;
/// enter your code below//
  tblbird.First;
  while not tblbird.eof do
    begin
      if pos('EAGLE',uppercase(tblbird['Birdname'])) <> 0 then
          reddisplay.Lines.Add(tblbird['Birdname']);
       tblbird.Next;
    end;

end;

//=============================================================================
==
//  Question 2.2.2   9 marks
//=============================================================================
==
procedure TQuestion_2.btnQuestion2_2_2Click(Sender: TObject);
var icount : integer;
begin
reddisplay.Clear;
/// enter your code below//
  icount := 0;
  tblbird.First;
  while not tblbird.eof do
    begin
      if tblbird['StatusID'] = 3 then
        begin
          tblbird.Edit;
          tblbird['StatusID'] := 2;
          tblbird.Post;
          inc(icount)
        end;
        tblbird.Next;
    end;
  reddisplay.Lines.Add('Number of changes made = ' + inttostr(icount));

end;
```

QUESTION 3

Class Unit:

```
unit Question3ClassDefinition;
interface
uses sysutils, dialogs, math;
type
tsighting = class (tobject)
private
  fname : string;
  farea : string;
  fbird : string;
  fdate : string;
  fquantity : integer;
public
constructor create(sbirder,sarea,sbirdname,sdateviewed : string);
procedure increasequantity(iqty : integer);
function sightinggap : string;
function tostring : string;
function getsightings : integer;
end;

implementation

{ tsighting }
     // 4 marks
constructor tsighting.create(sbirder, sarea, sbirdname, sdateviewed: string);
begin
  fname := sbirder;
  farea := sarea;
  fbird := sbirdname;
  fdate := sdateviewed;
  fquantity := 1;
end;

           //2 marks
function tsighting.getsightings: integer;
begin
result := fquantity;
end;
              // 4 marks
procedure tsighting.increasequantity(iqty: integer);
begin
fquantity := fquantity + iqty;
end;
```

```
            // 7 marks
function tsighting. sightinggap: string;
var stoday, syear: string;
begin
stoday := datetostr(date);
stoday := copy(stoday,1,4);

syear := fdate;
delete(syear,1,pos('/',syear));
delete(syear,1,pos('/',syear));
result := inttostr(strtoint(stoday) - strtoint(syear)) + ' years since first sighting';
end;
              // 5 marks
function tsighting.tostring: string;
begin
result :=
fname + ' first viewed a ' + fbird  + #13
+ sightinggap +  ' at ' + farea + ' on ' + fdate;
end;

end.
```

Main Unit:

```
/// provided code do not delete ///
var
  frmQuestion3: TfrmQuestion3;
  objbirder: tsighting;
implementation

{$R *.dfm}

procedure TfrmQuestion3.btnQ3_2_1Click(Sender: TObject); // 11 marks
var sname, sarea, sbird , sdate: string;
iday, imth, iyr : integer;
begin
sname := edtbirdername.text;
sarea := cmbarea.Text;
sbird :=  rdgbirds.Items[rdgbirds.ItemIndex];
sdate := inttostr(sedday.Value) + '/' + inttostr(sedMonth.Value) + '/' +
inttostr(sedyear.Value);
objbirder := tsighting.create(sname, sarea, sbird, sdate);
reddisplay.lines.add('');
reddisplay.lines.add(objbirder.tostring);

end;

procedure TfrmQuestion3.btnQ3_2_2Click(Sender: TObject); // 5 marks
begin
objbirder.increasequantity(sedsighting.value);
reddisplay.Lines.Add('Total sightings so far: ' + inttostr(objbirder.getsightings));

end;
```

QUESTION 4

```
var
  frmQuestion4: TfrmQuestion4;
   arrtypes : array[1..100] of string;
   arrqty : array[1..100] of integer;
   icount, isum : integer;
implementation
// Question 4 .1  19 marks
procedure TfrmQuestion4.btnQ4_1Click(Sender: TObject);
var
myfile : textfile;
soneline : string;
k,l : integer;
stemp : string;
itemp : integer;
begin
// provided code do not delete///
   reddisplay.Clear;
   reddisplay.Paragraph.TabCount := 1;
   reddisplay.Paragraph.Tab[0] := 100;
////////////////////////////////////////////
   icount := 0;
   isum := 0;
   assignfile(myfile, 'water.txt');
   reset(myfile);
   while not eof(myfile) do
     begin
       readln(myfile, soneline);
       inc(icount);
       arrtypes[icount] := copy(soneline,1, pos(',', soneline) - 1);
       delete(soneline,1, pos(',', soneline));
       arrqty[icount] := strtoint(soneline);
       isum := isum + arrqty[icount];
      end;

     for k := 1 to icount - 1  do
     for l := k + 1 to icount do
         if arrqty[k] > arrqty[l] then
           begin
             itemp := arrqty[k];
             arrqty[k] := arrqty[l];
             arrqty[l] := itemp;
             stemp := arrtypes[k];
             arrtypes[k] := arrtypes[l];
             arrtypes[l] := stemp;
           end;
   for k := 1 to icount do
   reddisplay.Lines.Add(arrtypes[k] + #9 + inttostr(arrqty[k]));
 reddisplay.Lines.Add('');
 reddisplay.Lines.Add('Total number of people = ' + inttostr(isum))

end;
```



```
 // Question 4 .2  13 marks
procedure TfrmQuestion4.btnQ4_2Click(Sender: TObject);
var iperc : integer;
arrperc : array[1..100] of integer;
k : integer;
sline : string;
begin
// provided code do not delete //////////////////////
 reddisplay.Clear;
   reddisplay.Paragraph.TabCount := 4;
   reddisplay.Paragraph.Tab[0] := 100;
    reddisplay.Paragraph.Tab[1] := 140;
    reddisplay.Paragraph.Tab[2] := 180;
    reddisplay.Paragraph.Tab[3] := 220;
   reddisplay.Lines.Add(#9 +'Percentage ranges');

   reddisplay.lines.add('Type' + #9 + '1-5' + #9 +  '6-10' + #9 + '11-20' + #9 + '21-50');
      reddisplay.Lines.Add('-------------------------------------------------------------------------------');
   ////////////////////////////////////////////////////////////////////

 for k := 1 to icount do
    begin
       iperc := round(arrqty[k]/isum *100);
       arrperc[k] := iperc;
     end;

  for k := 1 to icount do
  begin
  sline := arrtypes[k] + #9;
      case arrperc[k] of
        1..5 :  sline := sline + 'X';
        6..10 : sline := sline +#9 +  'X';
        11..20 : sline := sline +#9 + #9 +  'X';
        21..50 : sline := sline +#9 + #9 + #9 +  'X';
        end;
     reddisplay.lines.add(sline);
   end;

end;
```