Province of the
EASTERN CAPE
EDUCATION

NATIONAL SENIOR CERTIFICATE

GRADE 12

SEPTEMBER 2020

INFORMATION TECHNOLOGY P1 MARKING GUIDELINE

MARKS: 150

This marking guideline consists of 14 pages.

NAME OF LEARNER:				
TOTAL QUESTION 1:	**TOTAL QUESTION 2:**	**TOTAL QUESTION 3:**	**TOTAL QUESTION 4:**	**TOTAL**
/40	**/40**	**/40**	**/30**	**/150**

QUESTION 1: GENERAL PROGRAMMING SKILLS		MAX. MARKS	MARKS ACHIEVED
1.1	**Button [1.1 Display Club name]** Get name from edit box ✓ Join system date (converted to a string) ✓ to the end of the name ✓ Display name in panel ✓ Change font size of the panel to 24 ✓ Change font style of panel to bold ✓	**6**	
1.2	**Button [1.2 Process]** Get weight ✓as a number ✓ divide by 1000 ✓ remove decimals from weight (trunc or any other method) ✓ use global variables ✓ initialise the global variables ✓ Use case or IF ✓ Multiply correct amount per category ✓✓✓ Add to totals ✓ Display amounts✓ in correct panels ✓ for all three waste products ✓ formatted to currency ✓ 2 decimal places ✓	**16**	

1.3	**Button [1.3 Test for leap year]** Obtain the year from edit box ✓ convert to integer ✓ If year mod 400 = 0 ✓ then use message component to display leap year ✓ else ✓ If year mod 100 = 0 ✓ then use message component to display NOT leap year ✓ else ✓ if year mod 4 = 0 ✓ then use message component to display leap year ✓ else ✓ use message component to display NOT leap year ✓ *Alternative solution using a Boolean flag:* *Obtain the year from edit box ✓convert to integer ✓* *If year mod 400= 0 ✓* *then set Boolean variable to true ✓* *else ✓* *If year mod 100 = 0 ✓* *then set Boolean variable to false ✓* *else ✓* *if year mod 4 = 0 ✓* *then set Boolean variable to true ✓* *if Boolean flag is true* *use message component to display leap year* else ✓ *use message component to display NOT leap year ✓*	**12**	
1.4	**Button [1.4 Pattern]** Initialise string ✓ Initialise char ✓ Use a loop ✓ Skip 2 characters using any correct method ✓ Join third character to string ✓ Display line after the loop in the richedit ✓	**6**	
	TOTAL QUESTION 1	**40**	

QUESTION 2: OBJECT-ORIENTATED PROGRAMMING		MAX. MARKS	MARKS ACHIEVED
2.1.1	**Constructor Create:** Correct name ✓ with two string parameters ✓ Set fname and fleader to correct parameter values ✓✓ Set ftrees and fassistants to 0 ✓	**5**	
2.1.2	**Function calculatefunds : real** Correct method – function✓ Real data type✓ Calculation: ftrees x 2 ✓ + 250 ✓ + fassistants x 100 ✓ Return result ✓	**6**	
2.1.3	**Procedure addnumber;** Correct method – procedure ✓ Two integer parameters ✓ Add ✓ values to ftrees ✓ and fassistants ✓ Call calculatefunds ✓ to set ffunds attribute ✓	**7**	
2.1.4	**Function tostring: string;** Correct method – function ✓ String data type ✓ Compile string ✓ Correct attributes ✓ Convert fassistants and ftrees to integers ✓ Convert ffunds to currency and two decimals ✓ Use #13 correctly ✓ Return the string ✓	**8**	
	2.1 Subtotal: Object class	**26**	
2.2.1	**Button [Q2.2.1]** Instantiate the object Object name = ✓ tcountry.create ✓ With two string parameters ✓ In correct order ✓ Load picture into image component ✓ Enable the panel pnlQ2 ✓	**6**	
2.2.2	Get values from two spinedits ✓✓ Use both values as parameters ✓and the object ✓ to call procedure addnumbers ✓ Display in the label ✓using the object name ✓ and tostring function ✓	**8**	
	2.2 Subtotal: Form class	**14**	
	TOTAL QUESTION 2	**40**	

QUESTION 3: DATABASE PROGRAMMING		MAX. MARKS	MARKS ACHIEVED
3.1.1	**Button: [3.1.1]** SQL: select MaterialType from Material order by MaterialType DESC **Concepts:** SELECT correct field ✓ FROM correct table ✓ ORDER BY correct field ✓ DESC ✓	**4**	
3.1.2	**Button: [3.1.2]** SQL: Select MaterialType from Material where MaterialType like ' + quotedstr('%' + SLINE + '%') **Concepts:** SELECT correct field ✓ FROM correct table ✓ WHERE materialtype LIKE ✓ Quoted string (sline) ✓ Wildcards (%) in correct places ✓	**5**	
3.1.3	**Button: [3.1.3]** SQL: Select Datedisposed, MaterialType from Material where Month(DateDisposed) = 10 **Concepts:** SELECT two correct fields ✓ FROM correct table ✓ WHERE MONTH ✓ (datedisposed) ✓ Equal to 10 ✓	**5**	
3.1.4	**Button: [3.1.4]** SQL: update Disposal set Employees = Employees + Employees*10/100 **Concepts:** UPDATE Disposal ✓ Set Employees = ✓ Employees + ✓ Employees x 10% ✓ (also accept: employees x 1.1 or employees x 110/100)	**4**	
3.1.5	**Button: [3.1.5]** SQL: Select sum(Quantity) As Total_Quantities, DisposalMethod from Material, Disposal where Material.Disposalcode = Disposal.Disposalcode group by DisposalMethod **Concepts:** Select sum ✓ (Quantity) ✓ AS total_quantities ✓ Disposalmethod ✓ FROM Material, Disposal ✓ Where clause to join two tables ✓ Group by Disposalmethod ✓	**7**	
	3.1 Subtotal: SQL	**25**	

3.2.1	**Button: [3.2.1]** Move to first record of tblmaterial ✓ Loop while not end of table ✓ If disposalcode = 6 ✓ Then tblmaterial.edit ✓ Set greenhousefactor = 2 ✓ tblmaterial.post✓ Move to next record before end of loop ✓	**7**	
3.2.2	**Button: [3.2.2]** Move to first record of tblmaterial ✓ Loop while not end of table ✓ Running Total ✓of greenhousefactor ✓ x quantity ✓ Move to next record before end of loop ✓ Display total at end of edit box text ✓ Converted to real number ✓	**8**	
	3.2 Subtotal: Code constructs	**15**	
	TOTAL QUESTION 3	**40**	

QUESTION 4: PROBLEM SOLVING		MAX. MARKS	MARKS ACHIEVED
4.1	**Combobox option [Total weight of All waste]** Use case or nested IF for all options ✓ Loop using icount ✓ Add total to weights in array ✓ Display in panel converted to string after the loop ✓	**4**	
4.2	**Combobox option [Total weight of RECYCLED waste]** Initialise total variable ✓ Loop ✓ using icount If arrdata contains 'Recycled' ✓ Then Add total to weights in array ✓ Display in panel ✓ converted to string after the loop	**5**	
4.3	**Combobox option [Percentage of RECYCLED waste]** Recycled/total x 100 ✓ rounded ✓ Display in panel converted to string ✓	**3**	
4.4	**Combobox option [TOTAL weight of EACH waste product]** Initialise local totalling array ✓with 15 indices to zeros ✓ Loop from 1 to 15 (for arrtypes) ✓ Loop from 1 to icount (for arrdata) ✓ If ✓ arrdata (correct loop index) ✓ Contains arrtypes content (correct loop index) ✓ Then add weight ✓ from arrweights (correct loop index) ✓ to totalling array (correct loop index) ✓ assignfile statement for report.txt ✓ rewrite statement ✓ loop from 1 to 15 ✓ write to textfile ✓ a compiled string using arrtypes array ✓ and new total array ✓ converted to string ✓ closefile statement ✓	**18**	
	TOTAL QUESTION 4	**30**	

SAMPLE SOLUTIONS

QUESTION 1

```
var
frmQuestion1: TfrmQuestion1;
rcountpaper, rcountplastic, rcountglass : real;

implementation

procedure TfrmQuestion1.btnQ1_1Click(Sender: TObject);
begin
pnlclub.Caption := edtclub.Text + ' - ' + datetostr(date);
pnlclub.Font.Size := 24;
pnlclub.font.Style := [fsbold];
end;

procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject);
var
icode, inum : integer;
rweight : real;
begin
rweight := strtofloat(edtweight.Text)/1000;
rweight := trunc(rweight);

case rgpchoice.itemindex of
0 : rcountpaper := rcountpaper + rweight;
1 : rcountplastic := rcountplastic + rweight;
2 : rcountglass := rcountglass + rweight;
end;

pnlpaper.Caption := floattostrf(rcountpaper * 25,ffcurrency,10,2);
pnlplastic.Caption := floattostrf(rcountplastic  * 35,ffcurrency,10,2);
pnlglass.Caption := floattostrf(rcountglass * 40,ffcurrency,10,2);

end;

procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);
var iyear : integer;
begin
iyear := strtoint(edtyear.Text);
if (iyear mod 400 = 0) then showmessage(inttostr(iyear) + ' is a leap year')
  else
if iyear mod 100 = 0 then showmessage(inttostr(iyear) + ' is NOT a leap year')
  else
if (iyear mod 4 = 0) then showmessage(inttostr(iyear) + ' is a leap year')
  else
 showmessage(inttostr(iyear) + ' is NOT a leap year');
end;
```



```
procedure TfrmQuestion1.btnQ1_4Click(Sender: TObject);
var sline : string;
cold : char;
k : integer;
begin

sline := 'A';
cold := 'A';
for k := 1 to 8 do
  begin
    cold := succ(cold);
    cold := succ(cold);
     cold := succ(cold);
    sline := sline + cold;
  end;
  reddisplay.Lines.Add(sline);
end;
end.
```

QUESTION 2

Class Unit:

```
unit Question2ClassDefinition;

interface
/// provided code do not delete///
uses sysutils, dialogs, math;
type
Tcountry = class
private
  fcountry : string;
  fleader : string;
  ftrees : integer;
  ffunds : real;
  fassistants : integer;
public
constructor create(sname, sleader : string);
function calculatefunds : real;
procedure addnumbers(itrees, iassist : integer);
function tostring : string;
end;

implementation

constructor Tcountry.create(sname, sleader: string);
begin
  fcountry := sname;
  fleader := sleader;
  ftrees := 0;
 fassistants := 0;
end;
```

```
function Tcountry.calculatefunds: real;
begin
 result := ftrees * 2 +  250 + (fassistants * 100);
end;

procedure Tcountry.addnumbers(itrees, iassist: integer);
begin
  ftrees := ftrees + itrees;
  fassistants  := fassistants  + iassist;
  ffunds := calculatefunds;
end;

function Tcountry.tostring: string;
begin
result := fcountry + #13 +
      fleader + ' and ' + inttostr(fassistants) + ' assistants' + #13 +
      inttostr(ftrees) + ' trees' + #13 +
      'Funds: ' + floattostrf(ffunds,ffcurrency,10,2);
end;

end.
```

Main Unit:

```
var
  frmQuestion2: TfrmQuestion2;
  objcountry : tcountry;
implementation
{$R *.dfm}

procedure TfrmQuestion2.btnQ2_2_1Click(Sender: TObject);
begin
  objcountry := tcountry.create(edtcountry.Text, edtleader.Text);
  imgtrees.Picture.LoadFromFile('Trees.jpg');
  pnlQ2.Enabled := true;
end;

procedure TfrmQuestion2.btnQ2_2_2Click(Sender: TObject);
var itrees, ivol : integer;
begin
itrees := sedtrees.value;
ivol := sedassistants.Value;
objcountry.addnumbers(itrees, ivol);
lbldisplay.Caption := objcountry.tostring;
end;
```


QUESTION 3

```
//=============================================================================
//  Question 3.1.1
//=============================================================================
procedure TQuestion_3.btnQuestion3_1_1Click(Sender: TObject);
var
  sSQL1: String;
begin
  sSQL1 := 'select MaterialType from Material order by MaterialType DESC';
  // Provided code - do not change
  dbCONN.runSQL(sSQL1);
end;

//=============================================================================
//  Question 3.1.2
//=============================================================================
procedure TQuestion_3.btnQuestion3_1_2Click(Sender: TObject);
// Provided code - do not change/
var
  sline : string;
  sSQL2: String;
begin
  // Provided code - do not change//////////////////////
  sline := inputbox('Enter a Material','','wood');

  sSQL2 := 'Select MaterialType from Material where MaterialType like '  + quotedstr('%' +
SLINE + '%');

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

//=============================================================================
//  Question 3.1.3
//=============================================================================
procedure TQuestion_3.btnQuestion3_1_3Click(Sender: TObject);
// Provided code - do not change
var
  sSQL3: String;
begin

  sSQL3 := 'Select Datedisposed, MaterialType from Material where Month(DateDisposed) =
10';

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



```
//=====================================================================
//  Question 3.1.4
//=====================================================================
procedure TQuestion_3.btnQuestion3_1_4Click(Sender: TObject);
// Provided code - do not change
var
  sSQL4: String;
begin

  sSQL4 := 'update Disposal set Employees = Employees + Employees*10/100';

  // Provided code - do not change
  dbCONN.executeSQL(sSQL4,dbgdisposal,dbgmaterials,dbggarbage);
end;


//=====================================================================
//  Question 3.1.5
//=====================================================================
procedure TQuestion_3.btnQuestion3_1_5Click(Sender: TObject);
// Provided code - do not change
var
  sSQL5: String;
begin

  sSQL5 := 'Select sum(Quantity) As Total_Quantities, DisposalMethod from Material,
Disposal where Material.Disposalcode = Disposal.Disposalcode group by DisposalMethod' ;

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


//=====================================================================
//  Question 3.2.1
//=====================================================================
procedure TQuestion_3.btnQuestion3_2_1Click(Sender: TObject);
begin
/// enter your code below//
tblmaterial.First;
while not tblmaterial.eof do
begin
  if tblmaterial['Disposalcode'] = 6 then
  begin
   tblmaterial.edit;
   tblmaterial['Greenhousefactor'] := 2;
   tblmaterial.Post;
  end;
   tblmaterial.Next;
end;
end;
```



```
//=============================================================================
//  Question 3.2.2
//=============================================================================
procedure TQuestion_3.btnQuestion3_2_2Click(Sender: TObject);
var rtotal : real;
begin
/// enter your code below//
tblmaterial.First;
while not tblmaterial.eof do
begin
 rtotal := rtotal + tblmaterial['Greenhousefactor'] * tblmaterial['Quantity'];
  tblmaterial.Next;
end;
  edtdisplay.text := edtdisplay.text  + floattostr(rtotal);
end;
```


QUESTION 4

```
Const arrtypes : array[1..15] of string =
('Paper','Cardboard','Trash','Timber','Pallets','Rubber','Tyres','Metal','Food','Grass','Trees','Soil
','Rubble','Clay','Computers');
var
  frmQuestion4: TfrmQuestion4;
//provided code do not delete///////
   arrdata : array[1..100] of string;
   arrweights : array[1..100] of integer;
   icount : integer;

implementation {$R *.dfm}

procedure TfrmQuestion4.cmbreportChange(Sender: TObject);
var k, x, itotal, irecycled, ipos, ino : integer;
sline,sline1, sline2 : string;
tfile : textfile;
arrtotalweights: array[1..100] of integer;
inodup : integer;
icheck : integer;
bdup : boolean;
begin
 //Enter code below:
itotal := 0;
irecycled := 0;
for k := 1 to icount do
begin
 itotal := itotal + arrweights[k];
 if pos('RECYCLED' , uppercase(arrdata[k])) <> 0 then
    irecycled := irecycled + arrweights[k];
end;
case cmbreport.itemindex of
 0 : pnloutput.caption := inttostr(itotal);
 1 : pnloutput.caption := inttostr(irecycled);
 2 : pnloutput.caption := inttostr(round(irecycled/itotal*100));
 3 : begin
     for k := 1 to 100 do arrtotalweights[k] := 0;
     for x  := 1 to 15 do
     begin
      for k := 1 to icount do
       begin
         if pos(arrtypes[x],arrdata[k]) <> 0  then
           inc(arrtotalweights[x], arrweights[k]);
       end;
     end;
    assignfile(tfile, 'report.txt');
    rewrite(tfile);
    for k := 1 to 15 do
     writeln(tfile, arrtypes[k] + ' = ' + inttostr(arrtotalweights[k]));
    closefile(tfile);
    end;
end; // end of case    end;
```