Province of the
EASTERN CAPE
EDUCATION

NASIONALE SENIOR SERTIFIKAAT

GRAAD 12

SEPTEMBER 2020

INLIGTINGSTEGNOLOGIE V1
NASIENRIGLYN

PUNTE: **150**

Hierdie nasienriglyn bestaan uit 14 bladsye.

NAAM VAN LEERDER:				
TOTAAL VRAAG 1:	**TOTAAL VRAAG 2:**	**TOTAAL VRAAG 3:**	**TOTAAL VRAAG 4:**	**TOTAAL**
/40	**/40**	**/40**	**/30**	**/150**

VRAAG 1: ALGEMENE PROGRAMMERINGSVAARDIGHEDE		MAKS. PUNTE	PUNTE BEHAAL
1.1	**Knoppie [1.1 Vertoon Klubnaam]** Kry naam van redigeerblokkie ✓ Voeg stelseldatum (omgeskakel na string) ✓ aan einde van naam ✓ Vertoon naam in paneel ✓ Verander fontgrootte van paneel na 24 ✓ Verander die fontstyl van paneel na vetgedruk ✓	**6**	
1.2	**Knoppie [1.2 Prosesseer]** Kry gewig ✓as 'n syfer ✓ gedeel deur 1000 ✓ verwyder desimale van gewig (trunc of enige ander metode) ✓ gebruik globale veranderlikes ✓ inisialiseer die globale veranderlikes ✓ Gebruik case of IF ✓ Vermenigvuldig korrekte bedrag per kategorie ✓✓✓ Tel by totale ✓ Vertoon bedrae ✓ in korrekte panele ✓ Vir al drie afvalprodukte ✓ Geformateer as geldeenheid ✓ 2 desimale plekke ✓	**16**	

1.3	**Knoppie [1.3 Toets vir skrikkeljaar]** Kry die jaar van redigeerblokkie ✓ verander na integer ✓ As (if) jaar mod 400 = 0 ✓ Vertoon skrikkeljaar in afvoerkomponent ✓ else ✓ As (if) jaar mod 100 = 0 ✓ Vertoon NIE skrikkeljaar in afvoerkomponent ✓ else ✓ As (if) jaar mod 4 = 0 ✓ Vertoon skrikkeljaar in afvoerkomponent ✓ else ✓ Vertoon NIE skrikkeljaar in afvoerkomponent ✓ *Alternatiewe oplossing wat 'Boolean flag' gebruik:* *Kry die jaar van redigeerblokkie ✓verander na integer ✓* As (if) *jaar mod 400 = 0 ✓* *Stel Boolse veranderlike na true ✓* *else ✓* As (if) *jaar mod 100 = 0 ✓* *Stel Boolse veranderlike na false ✓* *else ✓* As (if) *jaar mod 4 = 0 ✓* *Stel Boolse veranderlike na true ✓* As (if) *Boolse veranderlike true is* *Vertoon skrikkeljaar in afvoerkomponent* *else ✓* *Vertoon NIE skrikkeljaar in afvoerkomponent ✓*	**12**	
1.4	**Knoppie [1.4 Gelukkige letters]** Inisialiseer string ✓ Inisialiseer char ✓ Gebruik 'n lus ✓ Gebruik enige korrekte metode om 2 karakters oor te slaan ✓ Voeg die derde karakter by die string ✓ Vertoon die reël in die richedit buite die lus ✓	**6**	
	TOTAAL VRAAG 1	**40**	

VRAAG 2: OBJEK-GEORIËNTEERDE PROGRAMMERING		MAKS. PUNTE	PUNTE BEHAAL
2.1.1	**Konstruktor Create:** Korrekte naam ✓ met twee string parameters ✓ Ken korrekte parameterwaardes toe aan fnaam en fleier ✓✓ Maak fbome en fassistente 0 ✓	**5**	
2.1.2	**Funksie BerekenKoste : real** Korrekte metode – funksie ✓ Real datatype ✓ Bereken: fbome x 2 ✓ + 250 ✓ + fassistente x 100 ✓ 'Return' result ✓	**6**	
2.1.3	**Prosedure TelBy;** Korrekte metode – prosedure ✓ Twee integer parameters ✓ Voeg waardes by ✓ fbome ✓ en fassistente ✓ Roep BerekenKoste ✓ om ffondse attribuut 'n waarde te gee ✓	**7**	
2.1.4	**Funksie tostring: string;** Korrekte metode – funksie ✓ String datatipe ✓ Skep 'n string ✓ Korrekte attribute ✓ Verander fassistente en fbome na string ✓ Verander ffondse na geldeenheid en twee desimale plekke ✓ Korrekte gebruik van #13 ✓ 'Return' die string ✓	**8**	
	2.1 Subtotaal: Objekklas	**26**	
2.2.1	**Knoppie [V2.2.1]** Instansieer die objek Objeknaam = ✓ tcountry.create ✓ Met twee string parameters ✓ In korrekte volgorde ✓ Laai die prentjie in component ✓ Maak die paneel pnlQ2 sigbaar ✓	**6**	
2.2.2	Kry waardes van twee spinedits ✓✓ Gebruik albei waardes as parameters ✓en die objek ✓om die prosedure TelBy te roep ✓ Vertoon in die label ✓gebruik die objeknaam ✓ en tostring funksie ✓	**8**	
	2.2 Subtotaal: Vormklas	**14**	
	TOTAAL VRAAG 2	**40**	

VRAAG 3: DATABASISPROGRAMMERING		MAKS. PUNTE	PUNTE BEHAAL
3.1.1	**Knoppie: [3.1.1]** SQL: select TipeMateriaal from Materiaal order by TipeMateriaal DESC **Konsepte:** SELECT korrekte veld ✓ FROM korrekte tabel ✓ ORDER BY korrekte veld ✓ DESC ✓	**4**	
3.1.2	**Knoppie: [3.1.2]** SQL: Select TipeMateriaal from Materiaal where TipeMateriaal like ' + quotedstr('%' + SLINE + '%') **Konsepte:** SELECT korrekte veld ✓ FROM korrekte tabel ✓ WHERE materialtype LIKE ✓ Quoted string (sline) ✓ 'Wildcards' (%) in korrekte plekke ✓	**5**	
3.1.3	**Knoppie: [3.1.3]** SQL: Select DatumVerwerk, TipeMateriaal from Materiaal where Month(DatumVerwerk) = 10 **Konsepte:** SELECT twee korrekte velde ✓ FROM korrekte tabel ✓ WHERE MONTH ✓ (datumverwerk) ✓ Gelyk is aan 10 ✓	**5**	
3.1.4	**Knoppie: [3.1.4]** SQL: update Verwerk set Werknemers = Werknemers + Werknemers*10/100 **Konsepte:** UPDATE Verwerk ✓ Set Werknemers = ✓ Werknemers + ✓ Werknemers x 10% ✓ (aanvaar: werknemers x 1.1 of werknemers x 110/100)	**4**	
3.1.5	**Knoppie: [3.1.5]** SQL: Select sum(Hoeveelheid) As Totale_Hoeveelhede, VerwerkMetode from Materiaal, Verwerk where Materiaal.VerwerkKode = Verwerk.VerwerkKode group by VerwerkMetode **Konsepte:** Select sum ✓ (Hoeveelheid) ✓ AS Totale_Hoeveelhede ✓ Verwerkmetode ✓ FROM Materiaal, Verwerk ✓ Where om twee tabelle te verbind ✓ Group by Verwerkmetode ✓	**7**	
	3.1 Subtotaal: SQL	**25**	

3.2.1	**Knoppie: [3.2.1]** Gaan na eerste rekord van tblmateriaal ✓ 'Loop while not end of table' ✓ If VerwerkKode = 6 ✓ dan tblmateriaal.edit ✓ Stel KweekhuisFaktor = 2 ✓ tblmateriaal.post ✓ Gaan na volgende rekord voor einde van lus ✓	**7**	
3.2.2	**Knoppie: [3.2.2]** Eerste rekord van tblmateriaal ✓ 'Loop while not end of table' ✓ 'running' Totaal ✓van kweekhuisFaktor ✓ x hoeveelheid ✓ Gaan na volgende rekord voor einde van lus ✓ Vertoon totaal aan die einde van die teks in die redigeerknoppie ✓ Omgeskakel na 'n 'real' getal ✓	**8**	
	3.2 Subtotaal: Kode-konstruktor	**15**	
	TOTAAL VRAAG 3	**40**	

VRAAG 4: PROBLEEMOPLOSSING		MAKS. PUNTE	PUNTE BEHAAL
4.1	**Kombinasieblokkie-opsie [Totale gewig van AL die afval]** Gebruik case of geneste IF vir al die opsies ✓ Lus wat icount gebruik ✓ Tel totale by gewigte in skikking ✓ Vertoon in paneel as 'n string buite die lus ✓	**4**	
4.2	**Kombinasieblokkie-opsie [Totale gewig van HERWONNE afval]** Inisialiseer totaal veranderlike ✓ Lus ✓ wat icount gebruik If arrdata 'Herwonne' bevat ✓ Dan voeg totaal by gewig in skikking ✓ Vertoon in paneel ✓ as 'n string buite die lus	**5**	
4.3	**Kombinasieblokkie-opsie [Persentasie van HERWONNE afval]** Herwonne/totaal x 100 ✓ afgerond ✓ Vertoon in paneel as 'n string ✓	**3**	
4.4	**Kombinasieblokkie-opsie [TOTALE gewig van ELKE afvalproduk]** Inisialiseer lokale skikking vir totale ✓met 15 indekse wat na nul gestel is ✓ Lus van 1 tot 15 (vir arrtipes) ✓ Lus van 1 na icount (vir arrdata) ✓ If ✓ arrdata (korrekte lusindeks) ✓ Bevat arrtipes inhoud (korrekte lusindeks) ✓ Tel gewig ✓ van arrgewig (korrekte lusindeks) ✓ by skikking vir totale (korrekte lusindeks) ✓ assignfile vir verslag.txt ✓ rewrite stelling✓ lus van 1 na 15 ✓ skryf na tekslêer ✓ 'n string wat uit arrtipes skikking ✓ en nuwe totaal skikking bestaan ✓ omgeskakel na string ✓ closefile stelling✓	**18**	
	TOTAAL VRAAG 4	**30**	

VOORGESTELDE OPLOSSINGS

VRAAG 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.
```

VRAAG 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;
```

VRAAG 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;
```

VRAAG 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;
```