



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

**NASIONALE
SENIOR SERTIFIKAAT**

GRAAD 12

INLIGTINGSTEGNOLOGIE V1

NOVEMBER 2023

NASIENRIGLYNE

PUNTE: 150

Hierdie nasienriglyne bestaan uit 24 bladsye.

ALGEMENE INLIGTING:

- Hierdie nasienriglyne moet as die basis vir die nasiensessie gebruik word. Dit is voorberei om deur nasieners gebruik te word. Daar word na alle nasieners verwag om 'n deeglike standaardiseringsvergadering by te woon om seker te maak dat die riglyne konsekwent geïnterpreteer en tydens die nasien van die kandidate se werk toegepas word.
- Let op dat leerders wat 'n alternatiewe korrekte oplossing as wat as voorbeeld van 'n oplossing in die nasienriglyne gegee word verskaf, volle krediet vir die relevante oplossing moet kry tensy die spesifieke instruksies in die vraestel nie gevolg is nie of die vereistes van die vraag nie nagekom is nie.
- **Bylaag A, B, C en D** (bladsy 3 tot 10) sluit die nasienrubriek vir elke om te gebruik vir enigeen van die twee programmeringstale in.
- **Bylaag E, F, G en H** (bladsy 11 tot 24) bevat voorbeelde in programmeringskode van oplossings vir VRAAG 1 tot VRAAG 4.
- Kopieë van **Bylaag A, B, C, D** en die **opsomming van die leerder se punte** (bladsy 3 tot 10) moet vir elke leerder gemaak word en tydens die nasiensessie voltooi word.

BYLAAG A**VRAAG 1: NASIENRUBRIEK – ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
1.1	<p>Knoppie [1.1 – Display name and age]</p> <p>Onttrek naam uit edtQ1_1 en stoor in sName veranderlike ✓ Onttrek ouderdom uit spnQ1_1 en stoor in iAge veranderlike ✓ Vertoon deur 'n afvoerdialoogblokkie te gebruik die naam ✓ met #13/#10/sLineBreak ✓ vir volgende reël die ouderdom, omgeskakel na 'n string ✓</p>	5	
1.2	<p>Knoppie [1.2 – Hockey teams]</p> <p>Onttrek die getal leerders uit die edit box, ✓ omgeskakel na 'n heelgetal/reële getal ✓ Bereken die aantal spanne Getal leerders DIV ✓ PLAYERS ✓ (gebruik konstante) Bereken die aantal reserwes Getal leerders MOD ✓ PLAYERS ✓ Vertoon die aantal spanne omgeskakel na String ✓ in die memoQ1_2 ✓ Vertoon ook die aantal reserwes ✓</p> <p>AANVAAR OOK: Alternatiewe wiskundige funksies wat die korrekte antwoord sal gee.</p> <p>LET WEL: Die gegewe konstante PLAYERS moet ten minste eenmaal gebruik word</p>	9	
1.3	<p>Knoppie [1.3 – Calculate]</p> <p>Formule: $d = \text{Sqrt} \left(\text{power} \left((rX - rY), 4 \right) \right)$ ✓ Vertoon die waarde van d in edtQ1_3 ✓ Geformateer met 3 desimale plekke ✓</p> <p>AANVAAR OOK: Alternatiewe wiskundige funksies wat die korrekte antwoord sal gee.</p> <p>MOENIE harde kodering aanvaar i.p.v. wiskundige funksies NIE.</p>	5	

1.4	<p>Knoppie [1.4 – Marathon results]</p> <p>case iPosisie of ✓ 1: ✓ Vertoon 'You receive a gold medal' ✓ 2, 3: Vertoon 'You receive a silver medal' ✓ 4..20: Vertoon 'You receive a bronze medal' ✓ Else Vertoon 'You receive a participation certificate' ✓ End // case</p> <p>NOTA: Die eerste twee punte is vir die struktuur van 'n gevalstelling, sal verbeur word wanneer veelvuldige keuse stelling gebruik word.</p>	6	
1.5	<p>Knoppie [1.5 – Average mark]</p> <p>Verklaar lêerveranderlike ✓ Inisialiseer Totaal en iTel na 0 ✓ AssignFile (tFile, 'Details.txt') ✓ Reset (tFile) ✓ Terwyl nie einde van lêer nie ✓ Begin Lees reel uit teksleer ✓ Inkrementeer iTel ✓ Vind die posisie van die # skeiteken ✓ Onttrek punt uit reel ✓ deur regte indekse te gebruik ✓ Skakel punt om na heel-/reële getal ✓ en voeg by Totaal ✓ Eindig Terwyl lus Eindig while Maak lêer toe Bereken gemiddeld deur Totaal en iTel te gebruik ✓ Afgerond tot die naaste heelgetal ✓ Vertoon gemiddelde punt in pnlQ1_5 ✓</p>	15	
TOTAAL AFDELING A:		40	

BYLAAG B**VRAAG 2: NASIENRUBRIEK – DATABASISPROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
2.1	SQL-stellings		
2.1.1	Knoppie [2.1.1 – Large enrolments] SELECT * ✓ FROM tblCourses ✓ WHERE MaxStudents > 99 ✓ Alternatief: MaxStudents >= 100	3	
2.1.2	Knoppie [2.1.2 – Lecturer gender] SELECT LecturerName, LecturerSurname, ✓ LEFT (Gender, 1) ✓ AS [Gender (M/F)] ✓ FROM tblLecturers ✓ Alternatief: MID (Gender, 1,1)	4	
2.1.3	Knoppie [2.1.3 – Multilingual lecturers] SELECT CourseID, CourseName FROM tblLecturers, tblCourses ✓ WHERE tblLecturers.LecturerID ✓ = tblCourses.LecturerID ✓ AND ✓ Multilingual = True ✓ ORDER BY CourseName ✓ Alternatiewe: Multilingual Multilingual LIKE True ORDER BY 2	6	
2.1.4	Knoppie [2.1.4 – Lecturer salaries] SELECT LecturerID, FORMAT(Count(*) ✓ * 10000 ✓, "CURRENCY") ✓ AS Salary FROM tblCourses ✓ GROUP BY LecturerID ✓ Let wel: Count kan enige veldnaam in plaas van * gebruik.	5	
2.1.5	Knoppie [2.1.5 – Change online option] UPDATE tblCourses ✓ SET OnlineOption = FALSE ✓ WHERE CourseName LIKE ✓ "%Programming%" ✓ Alternatief: CourseName LIKE "%Programming"	4	
	Subtotaal:	22	

VRAAG 2: NASIENRUBRIEK – VERVOLG

2.2	Databasismanipulasie		
2.2.1	Knoppie [2.2.1 – Average duration of courses]		
	<p>Gaan na die eerste rekord in tblLecturers ✓ Stap met lus ('loop') deur tblLecturers ✓ Vertoon opskrif deur LecturerID, LecturerName, en LecturerSurname in die regte formaat te gebruik ✓</p> <p>Inisialiseer Teller en Som veranderlikes ✓</p> <p>Gaan na die eerste rekord in tblCourses ✓ Stap met lus ('loop') deur tblCourses ✓ Toets of (tblLecturers ['LecturerID'] = tblCourses['LecturerID']) ✓ Inkrementeer Teller ✓ en Tel duration by Som ✓ Vertoon die Teller-waarde en kursus se naam ✓ tblCourses.Next ✓ Eindig lus (tblCourses)</p> <p>Bereken gemiddelde duur: Som / Teller ✓ Vertoon gemiddelde duur geformatteer na twee desimale ✓</p> <p>tblLecturers.Next ✓ Eindig lus (tblLecturers)</p>	14	
2.2.2	Knoppie [2.2.2 – Register new lecturer]		
	<p>tblLecturers.Insert; ✓ tblLecturers['LecturerID'] := 'ZT032'; tblLecturers['LecturerName'] := 'Zander'; tblLecturers['LecturerSurname'] := 'Thomas'; tblLecturers['Gender'] := 'Male'; tblLecturers['Multilingual'] := True; tblLecturers.Post; ✓</p> <p>Alternatiewe: Append i.p.v. Insert Enige navigeer-bevel in plaas van Post</p> <p>Toeken van veldwaardes: 1 punt vir die regte gebruik van al die veldname 1 punt vir die gebruik van regte waardes Trek 1 punt af vir elke fout tot 'n maksimum van twee punte</p>	4	
	Subtotaal:	18	
	TOTAAL AFDELING B:	40	

BYLAAG C**VRAAG 3: NASIENRUBRIEK – OBJEK-GEÖRIENTEERDE PROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
3.1.1	<p>Constructor Create</p> <p>Stel attribute (fSchoolName, fTotalLearners, fPublicSchool) ✓ Na regte parameters ✓ Ken 'Z' toe aan fRating ✓</p>	3	
3.1.2	<p>Function getPublicSchool</p> <p>Funksie-opskrif met Boolese waarde as terugstuurtype ✓ result = fPublicSchool ✓</p>	2	
3.1.3	<p>Procedure updateRating</p> <p>Prosedure-opskrif ✓ met heelgetalparameter ✓ slaagPersentasie = parameter / fTotalLearners *100 ✓ as slaagPersentasie >= 80 ✓ fRating = 'A' ✓ anders if slaagPersentasie >= 60 ✓ fRating = 'B' ✓ anders fRating = 'C' ✓</p> <p>Aanvaar ook ander oplossings</p> <p>Let wel: Die reeks 79 – 80 kan ook as 'n aparte reeks hanteer word – beide ingesluit of uitgesluit.</p>	8	
3.1.4	<p>Function calcFunding</p> <p>Funksie-opskrif met real as terugstuur-datatype ✓ Result ✓ = fTotalLearners ✓ * 145.50 ✓</p>	4	

3.1.5	<p>Function toString met string as terugstuur-datatype</p> <p>Bou string met fSchoolName en '-----' op volgende reel line ✓ Voeg 'Totale getal leerders: ' en fTotalLearners by die string ✓ Voeg 'Gradering: ' en fRating by die string ✓</p> <p>As fPublicSchool ✓ Voeg 'Publieke skool' by ✓ Anders Voeg 'Privaatskool' by ✓ Stuur string terug ✓</p>	7	
	Subtotaal: Objekklas	24	

VRAAG 3: NASIENRUBRIEK (VERVOLG)

VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER - PUNT
3.2.1	<p>Knoppie [3.2.1 – Instantiate Object]</p> <p>Onttrek skool se naam uit edtQ3_2_1 ✓ Onttrek getal leerders uit spnQ3_2_1 ✓ Onttrek publieke skool uit chbQ3_2_1 ✓</p> <p>objSchool ✓ := TSchool.create ✓ Gebruik drie argumente in regte volgorde ✓ (sSkoolNaam, iAantLeerders, bPubliekeSkool)</p> <p>Vertoon objek objSchool in redQ3 deur toString-metode te gebruik ✓</p>	7	
3.2.2	<p>Knoppie [3.2.2 – Rating]</p> <p>Onttrek getal leerders wat geslaag het uit spnQ3_2_2 ✓ Roep updateRating ✓ met regte argument ✓</p> <p>Vertoon objSchool in redQ3 deur toString-metode te gebruik ✓</p>	4	
3.2.3	<p>Knoppie [3.2.3 – Funding]</p> <p>Toets of getPublicSchool = TRUE ✓ Vertoon in redQ3 met boodskap Roep calcFunding-metode ✓ Geformateer in geldeenheid (currency) ✓</p> <p>Anders Vertoon boodskap – No funding available ✓</p>	5	
	Subtotaal Vormklas:	16	
	TOTAAL AFDELING C:	40	

BYLAAG D

VRAAG 4: NASIENRUBRIEK – PROBLEEMOPLOSSING

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
4.1	<p>Knoppie [4.1 – Codes]</p> <p>Lus ('loop') van 1 tot lengte van skikking (of 5) ✓ Inisialiseer sReel ← leë string ✓ Lus ('loop') van 1 tot ✓ die lengte van kode ✓ Toets of karakter in die kode ✓ 'n letter ('a'..'z' OR 'A'..'Z') ✓ of 'n syfer ('0'..'9') ✓ is Voeg die karakter by die sReel-afvoerstring ✓ Eindig binneste lus Bepaal die aantal spesiale karakters wat verwyder is Lengte(arrCodes[i]) ✓ – Lengte(sLine) ✓ // Of gebruik 'n teller in die binneste lus Voeg die sReel-kode ✓ in die list box in die regte format met hakies en aantal karakters wat verwyder is ✓ Eindig buitenste lus</p> <p>Konsepte:</p> <p>Buitenste i lus (1 tot 5) (1) Binneste j lus (1 tot lengte(arrCodes[i])) (2) Toets karakter [i][j] (1) as getal OF letter (2) Verwyder karakters / bou string (2) Tel karakters verwyder (2) Voeg nuwe kode by op IstQ4_1 (1) en aantal karakters verwyder (1)</p>	12	
4.2.1	<p>Knoppie [4.2.1 – Extra IT periods]</p> <p>Lus ('loop') iCnt van 1 tot 4 (Maandag – Donderdag) ✓ Stel teller op 1 ✓ Terwyl (die sel nie leeg is nie) ✓ Vermeerder teller met 1 ✓ Ken 'IT' ✓ toe aan arrTimeTable[iCnt, teller] ✓ Eindig lus</p> <p>Konsepte:</p> <p>Lus deur die dae van 1 tot 4 (1) // aanvaar ook 1 tot 5 Voorwaardelike lus/'Break'-stelling met for-lus (1) Bepaal indeks van (1) eerste leë spasie in 'n ry (1) Ken nuwe waarde aan leë spasie toe (2)</p>	6	

<p>4.2.2</p>	<p>Knoppie [4.2.2 – Group IT]</p> <p>Lus I van 1 tot 4 (Maandag – Donderdag) ✓ Inisialiseer Teller ✓ Lus J van 1 tot lengte van arrTimeTable[I] ✓ As arrTimeTable[I, J] = 'IT' ✓ As Teller = 1 ✓ Stoor indeks J (J_1) by eerste voorkoms van 'IT' ✓ As Teller = 2 ✓ Stoor indeks J (J_2) by tweede voorkoms van 'IT' ✓ // ruil ander vakkode met IT Stel sTemp ✓ na arrTimeTable[I, J_1+1] ✓ Stel arrTimeTable[I, J_1+1] na arrTimeTable[I, J_2] ✓ Stel arrTimeTable[I, J_2] na sTemp ✓ Eindig binneste lus Eindig buitenste lus</p> <p>Konsepte</p> <p>Lus deur rye (1 tot 4) //1 <i>Bepaal die posisie van die eerste voorkoms van IT:</i> Inisialiseer/skep Teller-veranderlike van 1^{ste} posisie//1 Lus deur die kolomme //1 Toets of die selwaarde = 'IT' //1 Stoor/bepaal die indeks/posisie van 1st voorkoms //2 <i>Bepaal die posisie van die tweede voorkoms van IT:</i> Stoor/bepaal die indeks/posisie van 2^{de} voorkoms //2</p> <p>Ruil die vakkode na die eerste voorkoms met die tweede voorkoms van IT //4</p>	<p>12</p>	
<p>TOTAAL AFDELING D:</p>		<p>30</p>	

OPSOMMING VAN LEERDER SE PUNTE:

<p>SENTRUMNUMMER:</p>		<p>LEERDER SE EKSAMENNUMMER:</p>			
	<p>AFDELING A</p>	<p>AFDELING B</p>	<p>AFDELING C</p>	<p>AFDELING D</p>	
	<p>VRAAG 1</p>	<p>VRAAG 2</p>	<p>VRAAG 3</p>	<p>VRAAG 4</p>	<p>GROOT-TOTAAL</p>
<p>MAKS. PUNTE</p>	<p>40</p>	<p>40</p>	<p>40</p>	<p>30</p>	<p>150</p>
<p>LEERDER SE PUNTE</p>					

BYLAAG E: OPLOSSING VIR VRAAG 1

```
unit Question1_u;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls, ExtCtrls, Spin, pngimage, Math;

type
  TfrmQuestion1 = class(TForm)
    grpQ1_2: TGroupBox;
    btnQ1_2: TButton;
    grpQ1_1: TGroupBox;
    edtQ1_1: TEdit;
    spnQ1_1: TSpinEdit;
    lblQ1_1_Name: TLabel;
    lblQ1_1_Age: TLabel;
    btnQ1_1: TButton;
    grpQ1_3: TGroupBox;
    imgQ1_3: TImage;
    btnQ1_3: TButton;
    edtQ1_3: TEdit;
    Label3: TLabel;
    grpQ1_5: TGroupBox;
    Label4: TLabel;
    Label5: TLabel;
    cmbQ1_5: TComboBox;
    btnQ1_5: TButton;
    grpQ1_: TGroupBox;
    btnQ1_4: TButton;
    pnlQ1_5: TPanel;
    Label6: TLabel;
    edtQ1_2: TEdit;
    memQ1_2: TMemo;
    lblQ1_4: TLabel;
    procedure btnQ1_1Click(Sender: TObject);
    procedure btnQ1_2Click(Sender: TObject);
    procedure btnQ1_3Click(Sender: TObject);
    procedure btnQ1_4Click(Sender: TObject);
    procedure btnQ1_5Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmQuestion1: TfrmQuestion1;

implementation

{$R *.dfm}
```

```
// =====  
// 1.1 Display name and age 5 marks  
// =====
```

```
procedure TfrmQuestion1.btnQ1_1Click(Sender: TObject);  
var  
    sName: String;  
    iAge: integer; // Provided code  
begin  
    sName := edtQ1_1.Text;  
    iAge := spnQ1_1.Value;  
    ShowMessage(sName + #13 + IntToStr(iAge));  
end;
```

```
// =====  
// 1.2 Hockey teams 9 marks  
// =====
```

```
procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject);  
const  
    PLAYERS = 11;  
var  
    iAantLeerdere, iAantSpanne, iAantReserve: integer;  
begin  
    // Provided code  
    memQ1_2.Clear;  
    //-----  
    iAantLeerdere := StrToInt(edtQ1_2.Text);  
    iAantSpanne := iAantLeerdere DIV PLAYERS;  
    iAantReserve := iAantLeerdere MOD PLAYERS;  
    memQ1_2.Lines.Add('Number of hockey teams: ' + IntToStr(iAantSpanne));  
    memQ1_2.Lines.Add('Number of learners on reserve list: ' + IntToStr  
        (iAantReserve));end;
```

```
// =====  
// 1.3 Calculate 5 marks  
// =====
```

```
procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);  
var  
    rX, rY: real; // Provided code  
    rD: real;  
begin  
    // Provided code  
    rX := 12.46;  
    rY := 8.54;  
  
    rD := sqrt(power((rX - rY), 4));  
    edtQ1_3.Text := FloatToStrF(rD, ffFixed, 8, 3);  
end;
```

```
// =====  
// 1.4 Marathon results 6 marks  
// =====  
  
procedure TfrmQuestion1.btnQ1_4Click(Sender: TObject);  
var  
    iPosisie: integer; // Provided code  
begin  
    // Provided code  
    iPosisie := StrToInt(TextBox('Marathon results',  
        'Enter the position the athlete achieved', '1'));  
    // -----  
    case iPosisie of  
        1:      lblQ1_4.Caption := 'You receive a gold medal.';  
        2, 3:   lblQ1_4.Caption := 'You receive a silver medal.';  
        4 .. 20: lblQ1_4.Caption := 'You receive a bronze medal.'  
        else lblQ1_4.Caption := 'You receive a participation certificate.';  
    end;  
end;  
  
// =====  
// 1.5 Average mark 15 marks  
// =====  
  
procedure TfrmQuestion1.btnQ1_5Click(Sender: TObject);  
var  
    tLeer: TextFile;  
    sReel: String;  
    iTotaal, iPunt, iTel, iPosHash: integer;  
    rGemid: real;  
begin  
    iTotaal := 0;  
    iTel := 0;  
    AssignFile(tLeer, 'Details.txt');  
    Reset(tLeer);  
    while NOT(eof(tLeer))do  
        begin  
            readln(tLeer, sReel);  
            iPosHash := pos('#', sReel);  
            iPunt := StrToInt(copy(sReel, iPosHash + 1));  
            iTotaal := iTotaal + iPunt;  
            inc(iTel);  
        end;  
    closeFile(tLeer);  
    rGemid := iTotaal / iTel;  
    pnlQ1_5.Caption := FloatToStrF(rGemid, ffFixed, 3, 0);  
end;  
  
end.
```

BYLAAG F: OPLOSSING VIR VRAAG 2

```
// =====  
// 2.1 - Afdeling: SQL-stellings  
// =====  
  
// =====  
// 2.1.1 Large courses 3 marks  
// =====  
  
    sSQL1 := 'SELECT * ' +  
            'FROM tblCourses ' +  
            'WHERE MaxStudents > 99';  
  
// =====  
// 2.1.2 Lecturer gender 4 marks  
// =====  
  
    sSQL2 := 'SELECT LecturerName, LecturerSurname, ' +  
            'Left(Gender, 1) AS [Gender (M/F)] ' +  
            'FROM tblLecturers';  
  
// =====  
// 2.1.3 Multilingual lecturers 6 marks  
// =====  
  
    sSQL3 := 'SELECT CourseID, CourseName ' +  
            'FROM tblLecturers , tblCourses ' +  
            'WHERE (tblLecturers.LecturerID = tblCourses.LecturerID) AND  
            (Multilingual = True) ' +  
            'ORDER BY CourseName';  
  
// =====  
// 2.1.4 Lecturer salaries 5 marks  
// =====  
  
    sSQL4 := 'SELECT LecturerID, ' +  
            'FORMAT(Count(*)*10000, "CURRENCY") ' +  
            'AS [Salary] ' +  
            'FROM tblCourses ' +  
            'GROUP BY LecturerID';  
  
// =====  
// 2.1.5 Change online option 4 marks  
// =====  
  
    sSQL5 := 'UPDATE tblCourses SET OnlineOption = FALSE ' +  
            'WHERE CourseName LIKE "%Programming%";
```

```
// =====  
// 2.2 - Afdeling: Delphi-kode  
// =====  
  
// =====  
// 2.2.1 Average duration of courses 14 marks  
// =====  
procedure TfrmQuestion2.btnQ2_2_1Click(Sender: TObject);  
var  
    iTelKursusse, iSomTydsduur: integer;  
    rGemTydsduur: real;  
begin  
    // Provided code  
    redQ2_2_1.Clear;  
  
    // 2.2.1 Average duration of courses  
  
    tblLecturers.First;  
    while NOT tblLecturers.Eof do  
    begin  
        redQ2_2_1.Lines.Add(tblLecturers['LecturerID'] + ': ' + tblLecturers  
            ['LecturerName'] + ' ' + tblLecturers['LecturerSurname']);  
        tblCourses.First;  
        iTelKursusse := 0;  
        iSomTydsduur := 0;  
        while NOT tblCourses.Eof do  
        begin  
            if tblLecturers['LecturerID'] = tblCourses['LecturerID'] then  
            begin  
                inc(iTelKursusse);  
                redQ2_2_1.Lines.Add(IntToStr(iTelKursusse) + '. ' + tblCourses  
                    ['CourseName']);  
                iSomTydsduur := iSomTydsduur + tblCourses['Duration'];  
            end;  
            tblCourses.Next;  
        end;  
        rGemTydsduur := iSomTydsduur / iTelKursusse;  
        redQ2_2_1.Lines.Add('Average duration of courses: ' + #9 + FloatToStrF  
            (rGemTydsduur, ffFixed, 8, 2) + #13);  
        tblLecturers.Next;  
    end;  
end;  
  
// =====  
// 2.2.2 Register new lecturer 4 marks  
// =====  
procedure TfrmQuestion2.btnQ2_2_2Click(Sender: TObject);  
begin  
    tblLecturers.Insert;  
    tblLecturers['LecturerID'] := 'ZT032';  
    tblLecturers['LecturerName'] := 'Zander';  
    tblLecturers['LecturerSurname'] := 'Thomas';  
    tblLecturers['Gender'] := 'Male';  
    tblLecturers['Multilingual'] := True;  
    tblLecturers.Post;  
end;
```

```
// =====  
// {$ENDREGION}  
// =====  
// {$REGION 'Provided code: Setup DB connections - DO NOT CHANGE!'}  
// =====  
  
procedure TfrmQuestion2.FormClose(Sender: TObject; var Action:  
TCloseAction);  
begin  
    // Disconnects from database and closes all open connections  
    dbCONN.dbDisconnect;  
end;  
  
procedure TfrmQuestion2.FormCreate(Sender: TObject);  
begin  
    redQ2_2_1.Paragraph.TabCount := 2;  
    redQ2_2_1.Paragraph.Tab[0] := 100;  
    redQ2_2_1.Paragraph.Tab[1] := 150;  
    redQ2_2_1.Paragraph.Tab[2] := 200;  
end;  
  
procedure TfrmQuestion2.FormShow(Sender: TObject);  
begin  
    // Sets up the connection to database and opens the tables.  
    dbCONN := TConnection.Create;  
    dbCONN.dbConnect;  
    tblLecturers := dbCONN.tblOne;  
    tblCourses := dbCONN.tblMany;  
    dbCONN.setupGrids(dbgLecturers, dbgCourses, dbgrdSQL);  
    pgcDBAdmin.ActivePageIndex := 0;  
end;  
// =====  
// {$ENDREGION}
```

end.

BYLAAG G: OPLOSSING VIR VRAAG 3**Objekklas:**

```

unit School_U;

interface

type
  TSchool = class(TObject)
  private
  var
    fSchoolName: String;
    fTotalLearners: Integer;
    fPublicSchool: boolean;
    fRating: char;
  public
    // Provide code
    constructor create(sSchoolName: String; iTotallLearners: integer;
      bPublicSchool: Boolean);
    // Code here

    function getPublicSchool: boolean;
    procedure updateRating(iLearnersPassed: integer);
    function calcFunding: real;
    function toString: String;
  end;

implementation

uses
  SysUtils, Math;
// =====
// 3.1.1 Constructor Create 3 marks
// =====

constructor TSchool.create(sSchoolName: String; iTotallLearners: integer;
  bPublicSchool: boolean);
begin
  // 3.1.1 Constructor Create
  fSchoolName := sSchoolName;
  fTotalLearners := iTotallLearners;
  fPublicSchool := bPublicSchool;
  fRating := 'Z';
end;
// =====
// 3.1.2 Function getPublicSchool 2 marks
// =====

function TSchool.getPublicSchool: boolean;
begin
  Result := fPublicSchool;
end;

```

```
// =====  
// 3.1.3 Procedure updateRating 8 marks  
// =====
```

```
procedure TSchool.updateRating(iLeerdersGeslaag: integer);  
var  
    rSlaagPer: real;  
begin  
    rSlaagPer := iLeerdersGeslaag / fTotalLearners * 100;  
  
    if rSlaagPer >= 80 then  
    begin  
        fRating := 'A';  
    end  
    else if (rSlaagPer >= 60) AND (rSlaagPer < 80) then  
    begin  
        fRating := 'B';  
    end  
    else  
    begin  
        fRating := 'C';  
    end;  
end;
```

```
// =====  
// 3.1.4 Function calcFunding 4 marks  
// =====
```

```
function TSchool.calcFunding: real;  
var  
    rFondse: real;  
begin  
    rFondse := 145.50 * fTotalLearners;  
    Result := rFondse;  
end;function TSchool.calcFunding: real;
```

```
// =====  
// 3.1.5 Function toString 7 marks  
// =====
```

```
function TSchool.toString: String;  
var  
    sAfvoerStr: String;  
begin  
    sAfvoerStr := fSchoolName + #13 + '-----' + #13;  
    sAfvoerStr := sAfvoerStr + 'Total number of learners: ' +  
        IntToStr(fTotalLearners) + #13;  
    sAfvoerStr := sAfvoerStr + 'Rating: ' + fRating + #13;  
    if fPublicSchool then  
        sAfvoerStr := sAfvoerStr + 'Public school ' + #13  
    else  
        sAfvoerStr := sAfvoerStr + 'Private school ' + #13;  
    Result := sAfvoerStr;  
end;  
  
end.
```

Hoofvormeenheid:

```
unit Question3_U;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls, CheckLst, ExtCtrls, Buttons, Spin, ComCtrls, jpeg;

type
  TfrmQuestion3 = class(TForm)
    gbxQ3_2_1: TGroupBox;
    gbxQ3_2_3: TGroupBox;
    redQ3: TRichEdit;
    btnQ3_2_1: TButton;
    gbxQ3_2_2: TGroupBox;
    btnQ3_2_2: TButton;
    Panel1: TPanel;
    Panel2: TPanel;
    btnQ3_2_3: TButton;
    Image1: TImage;
    Label6: TLabel;
    edtQ3_2_1: TEdit;
    Label2: TLabel;
    spnQ3_2_1: TSpinEdit;
    chbQ3_2_1: TCheckBox;
    Label1: TLabel;
    sedQ3_2_2: TSpinEdit;
    procedure btnQ3_2_1Click(Sender: TObject);
    procedure btnQ3_2_2Click(Sender: TObject);
    procedure btnQ3_2_3Click(Sender: TObject);
  private
  public
  end;

var
  frmQuestion3: TfrmQuestion3;

implementation

{$R *.dfm}

uses
  School_U;

var
  objSchool: TSchool;
```

```
// =====  
// 3.2.1 Instantiate object 7 marks  
// =====  
  
procedure TfrmQuestion3.btnQ3_2_1Click(Sender: TObject);  
var  
    sSchoolName : String;  
    iNumLearners : integer;  
    bPublicSchool : boolean;  
begin  
    // Provided code  
    redQ3.Clear;  
  
    // 3.2.1 Instantiate object  
    sSchoolName := edtQ3_2_1.Text;  
    iNumLearners := spnQ3_2_1.Value;  
    bPublicSchool := chbQ3_2_1.Checked;  
  
    objSchool := TSchool.create(sSchoolName, iNumLearners, bPublicSchool);  
    redQ3.Lines.Add(objSchool.toString);  
end;  
// =====  
// 3.2.2 Rating 4 marks  
// =====  
  
procedure TfrmQuestion3.btnQ3_2_2Click(Sender: TObject);  
var  
    iAantGeslaag: integer;  
begin  
    // Provided code  
    redQ3.Clear;  
  
    // 3.2.2 Rating  
    iAantGeslaag := spnQ3_2_2.Value;  
    objSchool.updateRating(iAantGeslaag);  
    redQ3.Lines.Add(objSchool.toString);  
end;  
  
// =====  
// 3.2.3 Funding 5 marks  
// =====  
  
procedure TfrmQuestion3.btnQ3_2_3Click(Sender: TObject);  
begin  
  
    // 3.2.3 Funding  
    if objSchool.getPublicSchool then  
        redQ3.Lines.Add('Public school will receive ' + FloatToStrF  
            (objSchool.calcFunding, ffCurrency, 8, 2))  
    else  
        redQ3.Lines.Add('No funding available ');  
end;  
  
end.
```

BYLAAG H: OPLOSSING VIR VRAAG 4

```

unit Question4_U;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics,
  Controls, Forms, Dialogs, StdCtrls, ComCtrls,
  ExtCtrls, jpeg, math;

type
  TfrmQuestion4 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    btnQ4_2_2: TButton;
    redQ4: TRichEdit;
    GroupBox1: TGroupBox;
    btnQ4_2_1: TButton;
    pgcQ4: TPageControl;
    tshQ4_1: TTabSheet;
    tshQ4_2: TTabSheet;
    btnQ4_1: TButton;
    lstQ4_1: TListBox;
    GroupBox2: TGroupBox;
    procedure btnQ4_2_2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure btnQ4_2_1Click(Sender: TObject);
    procedure btnQ4_1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
    procedure populate;
    procedure display;
  end;
var
  frmQuestion4: TfrmQuestion4;

  // Provided code for Question 4.1
  arrCodes: array [1 .. 5] of String =
    ('An7J*Q#D&N', 'pL78K#$.%BV', '89@FGh0&Y56#$Q', 'Bn4m321&*#T',
     'P2QwER%$#a');

  // Provided code for Question 4.2
  arrDays: array [1 .. 5] of String = ('Mon.', 'Tue.', 'Wed.', 'Thu.',
  'Fri. ');
  arrSubjectCodes: array [1 .. 5] of String =
    ('IT', 'HL', 'ACC', 'PHY', 'MAT' );
  arrTimeTable: array [1 .. 5, 1 .. 7] of String;

implementation

```

```
// =====  
// 4.1 Codes 12 marks  
// =====  
procedure TfrmQuestion4.btnQ4_1Click(Sender: TObject);  
var  
    I, J, iAantSpesKar: integer;  
    sReel: String;  
begin  
    // 4.1 Codes  
    for I := 1 to length(arrCodes) do  
    begin  
        sReel := '';  
        for J := 1 to length(arrCodes[I]) do  
        begin  
            if arrCodes[I][J] IN ['A' .. 'Z', 'a' .. 'z', '0' .. '9'] then  
            begin  
                sReel := sReel + arrCodes[I][J];  
            end;  
        end;  
        iAantSpesKar := length(arrCodes[I]) - length(sReel);  
        lstQ4_1.Items.Add(sReel + '(' + intToStr(iAantSpesKar) + ')');  
    end;  
end;
```

```
// =====  
// 4.2.1 Extra IT periods 6 marks  
// =====  
procedure TfrmQuestion4.btnQ4_2_1Click(Sender: TObject);  
var  
    iRy, iKol: integer;  
begin  
    // 4.2.1 Extra IT periods  
    for iRy := 1 to 4 do  
    begin  
        iKol := 1;  
        While NOT(arrTimeTable[iRy, iKol] = '') do  
        begin  
            inc(iKol);  
        end;  
        arrTimeTable[iRy, iKol] := 'IT';  
    end;  
    // Provided code  
    display;  
end;
```

```
// =====  
// 4.2.2 Group IT 12 marks  
// =====  
procedure TfrmQuestion4.btnQ4_2_2Click(Sender: TObject);  
var  
    I: integer;  
    J: integer;  
    iTel, iEerste, iTweede: integer;  
    sTemp: String;  
  
begin
```

```

// 4.2.2 Group IT
for I := 1 to 4 do
begin
  iTel := 0;
  for J := 1 to 7 do
  Begin
    if arrTimeTable[I, J] = 'IT' then
    begin
      inc(iTel);
      if iTel = 1 then
        iEerste := J + 1;
      if iTel = 2 then
      begin
        iTweede := J;
        sTemp := arrTimeTable[I, iEerste];
        arrTimeTable[I, iEerste] := arrTimeTable[I, iTweede];
        arrTimeTable[I, iTweede] := sTemp;
      end;
    end;
  end;
end;
// Provided code
display;
end;

// =====
// Provided code - Do not change
// =====

procedure TfrmQuestion4.populate;
var
  sSubjCode: String;
  iPeriod, iRand, iRow, iCol, iCnt: integer;
  arrLocal: array [1 .. 5] of String;
begin
  for iCnt := 1 to 5 do
  begin
    repeat
      iRand := RandomRange(1, 6);
      if length(arrLocal[iRand]) = 0 then
        arrLocal[iCnt] := arrSubjectCodes[iCnt];
    until length(arrLocal[iCnt]) > 0;
  end;
  for iCol := 1 to 5 do
  begin
    for iRow := 1 to 5 do
    begin
      repeat
        iRand := RandomRange(1, 8);
        until (arrTimeTable[iRow, iRand] = '');
        arrTimeTable[iRow, iRand] := arrLocal[iCol];
      end;
    end;
  end;
  display;
end;

```

```
procedure TfrmQuestion4.FormShow(Sender: TObject);
begin
  redQ4.Paragraph.TabCount := 9;
  redQ4.Paragraph.Tab[0] := 50;
  redQ4.Paragraph.Tab[1] := 100;
  redQ4.Paragraph.Tab[2] := 150;
  redQ4.Paragraph.Tab[3] := 200;
  redQ4.Paragraph.Tab[4] := 250;
  redQ4.Paragraph.Tab[5] := 300;
  redQ4.Paragraph.Tab[6] := 350;
  redQ4.Paragraph.Tab[7] := 400;
  redQ4.Paragraph.Tab[8] := 450;
  display;
  populate;
end;

procedure TfrmQuestion4.display;
var
  iRow, iCol, iCnt: integer;
  sLine: String;
begin
  sLine := #9;
  for iCnt := 1 to 7 do
    sLine := sLine + intToStr(iCnt) + #9;
  redQ4.Clear;
  redQ4.Lines.Add(sLine);
  for iRow := 1 to 5 do
    begin
      sLine := arrDays[iRow];
      for iCol := 1 to 7 do
        begin
          sLine := sLine + #9 + arrTimeTable[iRow, iCol];
        end;
      redQ4.Lines.Add(sLine);
    end;
  end;
end;

// =====
// End of provided code
// =====

end.
```