



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NASIONALE SENIOR SERTIFIKAAT

GRAAD 12

INLIGTINGSTEGNOLOGIE V1

NOVEMBER 2011

MEMORANDUM

PUNTE: 120

Hierdie memorandum bestaan uit 30 bladsye.

ALGEMENE INLIGTING:

- **Bladsye 2–11 bevat die Delphi-memoranda met moontlike oplossings vir VRAAG 1 tot 3 in programmeringskode.**
- **Bladsye 12–22 bevat die Java-memoranda met moontlike oplossings vir VRAAG 1 tot 3 in programmeringskode.**
- **Bladsye 23–30 bevat BYLAE A tot F wat 'n nasienblad vir elke vraag insluit vir kandidate wat enigeen van die twee programmeringstale gebruik.**
- **Kopieë van die toepaslike BYLAE behoort vir elke leerder gemaak te word om tydens die nasiensessie te voltooi.**

AFDELING A: DELPHI**VRAAG 1: PROGRAMMERING EN DATABASIS**

```
unit Vraag1_U;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ADODB, Grids, DBGrids, ExtCtrls, Buttons;

type
  TfrmVraag1 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    btnA: TButton;
    btnB: TButton;
    btnC: TButton;
    btnD: TButton;
    btnE: TButton;
    btnF: TButton;
    btnG: TButton;
    BitBtn1: TBitBtn;
    qryRec: TADOQuery;
    tblRecAg: TDataSource;
    grdRec: TDBGrid;

    procedure btnAClick(Sender: TObject);
    procedure btnBClick(Sender: TObject);
    procedure btnCClick(Sender: TObject);
    procedure btnDClick(Sender: TObject);
    procedure btnEClick(Sender: TObject);
    procedure btnFClick(Sender: TObject);
    procedure btnGClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmVraag1: TfrmVraag1;
implementation

{$R *.dfm}
```

Sien BYLAE A vir alternatiewe kode en nasienriglyne

```

procedure TfrmVraag1.btnAClick(Sender: TObject);
begin
    qryRec.Active := False;                                     // VRAAG 1.1
    qryRec.SQL.Text := 'SELECT *✓ FROM tblDamme✓ ORDER BY HoogteVanWal✓ ASC';
    qryRec.Active := True;
end;                                                            (3)
//=====
procedure TfrmVraag1.btnBClick(Sender: TObject);               // VRAAG 1.2
var
    pr : String;
begin
    qryRec.Active := False;
    pr := InputBox('Groot Dorpe', 'Tik die naam van die provinsie in', ''); ✓
    qryRec.SQL.Text := 'SELECT DorpNaam, Bevolking✓ FROM tblDorpe WHERE✓
        Bevolking > 100000✓ AND✓ Provinsie = "' + pr + '"';
    qryRec.Active := True;
end;                                                            (6)
//=====
procedure TfrmVraag1.btnCClick(Sender: TObject);              // VRAAG 1.3
begin
    qryRec.Active := False;
    qryRec.SQL.Text := 'SELECT DamID, DamNaam, ✓ (YEAR(NOW())✓ - JaarVoltooi✓) AS
        Ouderdom✓, ROUND(DamVlak / Kapasiteit * 100✓, 1✓) AS Persentasie✓ FROM
        tblDamme';
    qryRec.Active := True;
end;                                                            (7)
//=====
procedure TfrmVraag1.btnDClick(Sender: TObject);              // VRAAG 1.4
begin
    qryRec.Active := False;
    qryRec.SQL.Text := 'SELECT Provinsie✓, COUNT(*)✓ AS KritiekeDorpe✓ FROM
        tblDorpe WHERE WaterBeperkings = TRUE ✓GROUP BY Provinsie✓';
    qryRec.Active := True;
end;                                                            (5)
//=====
procedure TfrmVraag1.btnEClick(Sender: TObject);              // VRAAG 1.5
begin
    qryRec.Active := False;
    qryRec.SQL.Text := 'SELECT DISTINCT Provinsie✓ FROM tblDorpe✓, tblDamme✓
        WHERE tblDorpe.DamID✓ = tblDamme.DamID✓ AND Rivier✓ = "Vaalrivier"✓';
    qryRec.Active := True;
end;                                                            (7)
//=====
procedure TfrmVraag1.btnFClick(Sender: TObject);              // VRAAG 1.6
begin
    qryRec.Active := False;
    qryRec.SQL.Text := 'UPDATE tblDorpe✓ SET✓ WaterBeperkings = True✓ WHERE
        Provinsie = "Noordwes" ✓';
    qryRec.ExecSQL;
    MessageDlg('Rekords Suksesvol Verwerk',mtInformation,[mbok],0);
end;                                                            (4)
//=====
procedure TfrmVraag1.btnGClick(Sender: TObject);              // VRAAG 1.7
begin
    qryRec.Active := False;
    qryRec.SQL.Text := 'DELETE✓ FROM tblDamme✓ WHERE HoogteVanWal < 11.50✓';
    MessageDlg('Rekords Suksesvol Verwerk',mtInformation,[mbok],0);
    qryRec.ExecSQL;
end;                                                            (3)
//=====
end.                                                            [35]

```

VRAAG 2: OBJEK-GEORIËNTEERDE PROGRAMMERING**unit uHuishoudingXXXX;**

```

interface
uses SysUtils;
  type
    arrTipe = array[1..7] of integer;
    THuishouding = class (TObject)
    private
      fRekNom :string;
      fLede :integer;
      fWaterG :arrTipe;
    public
      constructor create(aRekNom : string; aLede :integer;arrWaterG :
                          arrTipe );

      function berekenTotaal:integer;
      function berekenGem:double;
      function bepaalHoogsteDag:integer;
      function bepaalHoogRisiko(dagLimiet:real):boolean;
      function toString:string;

    end;

```

implementation

```

//=====
// V 2.1.1 (3)

```

```

  constructor THuishouding.create(aReknom : string; aLede:integer;
                                arrWaterG:arrTipe);

```

begin

```

  fRekNom := aRekNom; ✓
  fLede := aLede; ✓
  fWaterG := arrWaterG; ✓

```

end;

V 2.1.1

(3) Ken parameters toe aan privaat velde

```

  Aanvaar 'n lus om aan skikking toe te ken
  Trek net een punt af as die toekenningstellings verkeerd-om is bv.
  aReknom := fReknom

```

```

//=====
// V 2.1.2 (4)

```

```

  Ignoreer enige foute in die definisie(verklaring)van metode - tel geen punte
  Terugstuurtipe kan int of double wees

```

function THuishouding.berekenTotaal:integer;

var

iTotaal, k :integer;

begin

```

  iTotaal := 0; ✓
  for k := 1 to length(fWaterG) do ✓
    iTotaal := iTotaal + fWaterG[k]; ✓
    // of inc(iTotaal, fWaterG[k]);
  result := iTotaal; ✓

```

end;

V 2.1.2

(1) Inisialiseer totaal
 (1) for lus
 (1) Voeg skikkingselement by totaal
 (1) stuur totaal terug (gebruik result of die naam van die funksie)

Aanvaar: iTotaal as 'n globale veranderlike.**Aanvaar: lus tot <=7 of < 8****Aanvaar: Optel van skikkingwaardes individueel - geen lus****Aanvaar: gebruik nie lokale iTotaal veranderlike - tel op en ken toe aan****Result - alles in een**

Ken 4 punte toe as metode/kode reg is maar in die hoofteenheid is

//=====

// V 2.1.3 (2)

```
function THuishouding.berekenGem:real; ✓
begin
    result := berekenTotaal / 7; ✓
end;
```

V 2.1.3

- (1) Datatipe van die terugstuurwaarde is real (of double)
- (1) Regte berekening

Aanvaar die gebruik van iTotaal slegs as berekenTotaal geroep is (kan in die hoofteenheid geroep word)
Aanvaar as die totaal weer hier bereken word.
Aanvaar integer as 'n terugstuurtype.

Ken 2 punte toe as metode/kode reg is maar in die hoofteenheid is

//=====

// V 2.1.4 (8/2 = 4) (na bo afgerond)

```
function THuishouding.bepaalHoogsteDag:integer; ✓
var
    iHoogsteDag, iHoogste, k :integer;
begin
    iHoogsteDag := 1; ✓
    iHoogste := fWaterG[1]; ✓
    for k := 2 to 7 do ✓
        begin
            if (fWaterG[k] > iHoogste) ✓ then
                begin
                    iHoogsteDag := k; ✓
                    iHoogste := fWaterG[k]; ✓
                end;
            result := iHoogsteDag; ✓
        end;
    end;
```

V 2.1.4

- (1) Stuur heelgetal ('integer') terug
- (1) Inisialiseer iHoogsteDag
- (1) Inisialiseer iHoogste
- (1) For lus
- (1) if-stelling
- (1) Verander iHoogsteDag
- (1) Verander iHoogste
- (1) Stuur iHoogsteDag terug

Aanvaar sortering van hoeveelhede, as regte dag ook teruggestuur word - vol punte
Aanvaar variasies wat reg is om die hoogste te kry bv. begin met 0 as hoogste ipv die eerste element.
Sortering reg gedoen maar nie die regte dag gekry nie - 3 uit 4 punte

Ken 4 punte toe as metode/kode reg is maar in die hoofteenheid is

//=====

// V 2.1.5 (9)

```
function THuishouding.bepaalHoogRisiko(dagLimiet:real):boolean;
var
    rGem      :real;
    iTel,k    :integer;
begin
    rGem := berekenGem;
    iTel := 0; ✓
    for k := 1 to length(fWaterG) do ✓
        begin
            if (fWaterG[k] > dagLimiet) then ✓
                inc(iTel); ✓
        end;
    if ((rGem > dagLimiet) ✓ OR (iTel > 2)) ✓ then
        result := true ✓
```

V 2.1.5

- (1) Inisialiseer iTel
- (1) Lus
- (1) if skikkingselement > dagLimiet
- (1) Inkrementeer teller
- (3) if rGem > dagLimiet or iTel > 2
- (1) Stuur true terug
- (1) else stuur false terug

```

    else
        result := false; ✓
end;

```

```

Aanvaar globale veranderlikes
Moenie punte aftrek vir toevoer van dagLimiet
Aanvaar: if (berekenGem() > dagLimiet ✓ || ✓iTel > 2) ✓
Aanvaar: 'n Enkele stelling wat 'n Boolean waarde terugstuur
Result = ✓ (rGem > dagLimiet ✓ OR ✓iTel > 2) ✓✓
Aanvaar: Initialiseer 'n Boolean veranderlike, stuur Boolean veranderlike
terug

```

```

//=====
// V 2.1.6 (6)

```

```

1 punt vir elke stukkie inligting = 5 punte
1 punt om al die inligting saam te voeg in een string

```

```

function THuishouding.toString:string;
var
    sObjStr:string;
    k      :integer;
begin
    sObjStr := 'Rekeningnommer: ' + fRekNom + #13 + 'Getal lede: ' +
                IntToStr(fLede) + #13;
    sObjStr := sObjStr + 'Daaglikse waterverbruik' + #13 ✓
    for k := 1 to 7 do
        sObjStr := sObjStr + intToStr(k) ✓ + #9;

        sObjStr := sObjStr + #13 + 'Waterverbruik:' ✓ + #9;
        for k := 1 to length(fWaterG) do ✓
            sObjStr := sObjStr + IntToStr(fWaterG[k]) ✓ + #9;
            ✓ Voeg stringe saam

        result := sObjStr;
end;

```

V 2.1.6
 (1) Opskrif + nuwe reel(#13 of #10)
 (1) Dagnommers
 (1) Opskrif
 (2) Waardes van skikking
 (1) Las stringe aanmekaar

```

Aanvaar aparte elemente van skikking ipv die lus.
Aanvaar enige korrekte vorm van saamvoeg van al die inligting in 'n string

```

```

//=====
unit Vraag2XXXX_U;

```

```

interface

```

```

uses

```

```

    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, Menus, StdCtrls, ComCtrls;

```

```

type

```

```

    TfrmVraag2 = class(TForm)
        MainMenu1: TMainMenu;
        mnuOptionB: TMenuItem;
        redAfvoer: TRichEdit;
        mnuOptionC: TMenuItem;
        mnuOptionA: TMenuItem;
        Quit1: TMenuItem;
        procedure mnuOptionBClick(Sender: TObject);
        procedure FormActivate(Sender: TObject);
        procedure mnuOptionCClick(Sender: TObject);
        procedure mnuOptionAClick(Sender: TObject);
        procedure Quit1Click(Sender: TObject);

```

```

    private

```

```

public
{ Public declarations }
end;

var
    frmVraag2: TfrmVraag2;

implementation
uses
    uHuishoudingXXXX;
//=====

```

// V 2.2.1**(2)**

```

var
    Huishouding :THuishouding; ✓
    sReknom      :string;
    iLede        :integer;
    arrWaterG    :arrTipe = (481, 438, 454, 353, 421, 396, 432);

```

V 2.2.1**(2) Verklaar die objekveranderlike**

```

{$R *.dfm}
procedure TfrmVraag2.FormActivate(Sender: TObject);
begin
    sRekNom := 'AC-23245';
    iLede   := 4;
    Huishouding := THuishouding.create(sRekNom, iLede, arrWaterG); ✓
end;

```

Trek 1 punt af indien geen parameters.

```

procedure TfrmVraag2.Quit1Click(Sender: TObject);
begin
    Application.Terminate;
end;
//=====

```

// V 2.2.2**(4)**

```

procedure TfrmVraag2.mnuOptionAClick(Sender: TObject);
begin
    redAfvoer.Clear;
    redAfvoer.Lines.Add(Huishouding.toString); ✓
    redAfvoer.Lines.Add(' ');
    redAfvoer.Lines.Add('Totale waterverbruik: '✓+
        IntToStr(Huishouding.berekenTotaal) ✓ + ' liter');
    redAfvoer.Lines.Add('Gemiddelde waterverbruik per dag: ' +
        FloatToStrF(Huishouding.berekenGem, ✓ ffFixed,8,1) + ' liter');
end;

```

V 2.2.2

(1) Roep die toString metode van die objek
 (1) Vertoon byskrif
 (1) Roep die berekenTotaal metode
 (1) Roep die berekenGem-metode

Moenie streng wees met die bewoording van byskrifte of formattering van waardes nie

```

//=====
// V 2.2.3
(6)

```

```

procedure TfrmVraag2.mnuOptionBClick(Sender: TObject);
var
    rGem:real;
    k    :integer;
begin
    redAfvoer.Clear;
    rGem := Huishouding.berekenGem; ✓
    redAfvoer.Lines.Add('Dae en die hoeveelheid water waarmee die gemiddeld
        oorskry word');
    redAfvoer.Lines.Add('=====');

```

V 2.2.3

(1) Roep berekenGem-metode
 (1) Vertoon gemiddeld
 (1) Lus
 (1) if
 (2) Vertoon getal & verskil

```

redAfvoer.Lines.Add('Gemiddelde waterverbruik per dag: ' +
    FloatToStrF(Huishouding.berekenGem, ffFixed, 8, 1)✓ + ' liter');
redAfvoer.Lines.Add('Dae      Waarde oorskry gemiddeld met (liter)');
for k := 1 to length(arrWaterG) do✓
begin
    if (arrWaterG[k] > rGem) then✓
    begin
        redAfvoer.Lines.Add(IntToStr(k)✓ + #9 + FloatToStrF(arrWaterG[k]-
            rGem,✓ ffFixed,8,1));
    end;
end;
end;

```

Geen punte vir opskrifte nie
 Vertoon gemiddeld – Maak nie saak hoe die gemiddeld verkry is nie, punt is
 nie vir formattering nie
 Vierde punt is vir berekening, nie vir formattering nie

//=====

// V 2.2.4

(5)

```

procedure TfrmVraag2.mnuOptionCClick(Sender: TObject);
var
    rDagLimiet :real;
begin
    redAfvoer.Clear;
    rDagLimiet := StrToFloat(InputBox('Waterlimiet',
        'Tik die limiet water per dag in', ''));✓
    redAfvoer.Lines.Add(Huishouding.toString);✓
    redAfvoer.Lines.Add('');
    redAfvoer.Lines.Add('Die dag waarop die meeste water verbruik is: ' +
        intToStr(Huishouding.bepaalHoogsteDag));✓

    if (Huishouding.bepaalHoogRisiko(rDagLimiet))✓ then
        redAfvoer.Lines.Add('Hoërisiko-huishouding')
    else
        redAfvoer.Lines.Add('Nie 'n hoërisiko-huishouding nie'); } ✓
end;
end.

```

V 2.2.4

- (1) Tik rDaglimiet in
- (1) Roep toString
- (1) Roep bepaalHoogsteDag()
- (1) If-stelling
- (1) Vertoon regte boodskap

rDagLimiet – integer of real
 Tweede punt: Vir roep van toString – geen ander manier aanvaar om te vertoon
 Derde punt gaan vir roep van metode, nie byskrif. Aanvaar geen byskrif
 Vierde punt: vir roep van metode as deel van 'n if of toekenning aan
 veranderlike
 Vyfde punt: Vertoon van boodskap – punt vir twee boodskappe met else of
 tweede if

[45]

VRAAG 3: DELPHI-PROGRAMMERING

LET WEL: Hierdie is slegs 'n voorbeeld – leerders kan hierdie vraag beantwoord op enige manier wat hulle wil. Gebruik die algemene rubriek in die nasienblaaie vir nasien.

unit Vraag3_U;

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, ComCtrls, ExtCtrls, Buttons;
```

```
type
```

```
TfrmVraag3 = class(TForm)  
    redAfvoer: TRichEdit;  
    pnlButtons: TPanel;  
    btnA: TButton;  
    btnB: TButton;  
    BitBtn1: TBitBtn;  
    procedure btnAClick(Sender: TObject);  
    procedure btnBClick(Sender: TObject);
```

```
private
```

```
{ Private declarations }
```

```
public
```

```
{ Public declarations }
```

```
end;
```

```
var
```

```
frmVraag3: TfrmVraag3;  
iAantVerw : integer;  
arrVerw, arrInhoud : array[1..100] of String;
```

```
implementation
```

```
//=====
```

//VRAAG 3.1

```
procedure SkepVoorstelleLeer;
```

```
var
```

```
sLeer : TextFile;
```

```
begin
```

```
AssignFile(sLeer, 'Voorstelle.txt');
```

```
ReWrite(sLeer);
```

```
CloseFile(sLeer);
```

```
end;
```

```
//=====
```

//VRAAG 3.2**(6)**

```
function valideerRekNom(sReknom:String): boolean; ✓
```

```
var
```

```
bGeldig : boolean;
```

```
begin
```

```
bGeldig := false; ✓
```

```
if (length(sReknom) = 7) ✓ and (sReknom[1] in ['A'..'Z']) ✓ then
```

```
begin
```

```
bGeldig:= true; ✓
```

```
end;
```

```
result := bGeldig; ✓
```

```
end;
```

V 3.1

Kode is gegee in die Afrikaans Java weergawe
2 punte toegeken in Vraag 3.3

V 3.2

- (1) Opskrif van subprogram
- (1) Inisialiseer Boolese waarde
- (2) if stelling
- (1) Verander Boolese waarde
- (1) Stuur Boolese waarde terug

Aanvaar: if ... else in plaas van initialisering van Boolese waarde
Aanvaar: Enige regte kode om eerste karakter te kry
Aanvaar: Een stelling in metode wat die Boolese waarde terugstuur bv.
Result := (lenqth(sRekNom) and);

//=====

//VRAAG 3.3**(24) + 2**

```

procedure TfrmVraag3.btnAClick(Sender: TObject);
var
  inLeer, VoorstelLeer : textfile;
  sReel, sRekNom, sNavraag, sDatum, sTipe, sVerwNom : String;
  k, iKlagtes, iRekening : integer;
begin
  CreateSuggestionsFile; ✓
  AssignFile(inLeer, 'DataA.txt'); ✓
  Reset(inLeer); ✓
  AssignFile(VoorstelLeer, 'Voorstelle.txt');
  Append(VoorstelLeer); ✓

  iAantVerw := 0;
  iKlagtes := 1;
  iRekening := 1;

  while NOT EOF (inLeer) do ✓
    begin
      ReadLn(inLeer, sReel); ✓

      sTipe := Copy(sReel, 1, Pos(':', sReel) - 1); ✓
      Delete(sReel, 1, Pos(':', sReel));

      sReknom := Copy(sReel, 1, Pos(':', sReel) - 1); ✓
      Delete(sReel, 1, Pos(':', sReel));

      sDatum := Copy(sReel, 1, Pos('#', sReel) - 1); ✓
      Delete(sReel, 1, Pos('#', sReel));

      sNavraag := sReel; ✓

      if ValideerRekNom(sRekNom) then ✓
        begin
          if (sTipe = 'Voorstel') then ✓
            WriteLn(VoorstelLeer, sRekNom + ':' + sDatum + '#' + sNavraag); ✓
          else
            begin
              Inc(iAantVerw); ✓
              sVerwNom := sTipe[1]; ✓
              if (sTipe = 'Klagte') then ✓
                begin
                  sVerwNom := sVerwNom + IntToStr(iKlagtes); ✓
                  Inc(iKlagtes); ✓
                end
              else if (sTipe = 'Rekening') then
                begin
                  sVerwNom := sVerwNom + IntToStr(iRekening); ✓
                  Inc(iRekening); ✓
                end;
              sVerwNom := sVerwNom + '-' + sRekNom + '-' + sDatum; ✓
              arrVerw[iAantVerw] := sVerwNom; ✓
              arrInhoud[iAantVerw] := sNavraag; ✓
            end;
          end;
        end;

      redAfvoer.Lines.Clear;
      redAfvoer.Lines.Add('Verwysingsnommers');
      redAfvoer.Lines.Add('=====');

```

V 3.3

- (1) Skep Voorstelle-lêer
- (1) Maak Voorstelle lêer oop vir skryf
- (2) Maak Data lêer oop
- (1) While not eof
- (1) Lees 'n reël
- (1) Onttrek tipe kwessie
- (1) Onttrek rekeningnommer
- (1) Onttrek datum
- (1) Onttrek kwessie
- (1) Roep valideerReknom
- (1) Toets vir voorstel
- (1) Skryf voorstel na lêer
- (1) Binne else vermeerder verwysingsnommer se teller
- (1) Kry eerste letter van kwessie
- (1) Toets kategorie
- (2) Skep verwysingsnommer vir klagte
- (2) Skep verwysingsnommer vir rekening
- (1) Skep kwessie verwysingsnommer
- (1) Stoor verwysingsnommer in skikking
- (1) Stoor navraag in skikking
- (2) Vertoon Verwysingsnommers
- (1) Maak Voorstelle-lêer toe

```

for k := 1 to iAantVerw do✓
  begin
    redAfvoer.Lines.Add(arrVerw[k]); ✓
  end;
CloseFile(inLeer);
CloseFile(VoorstelLeer); ✓
end;

```

Aanvaar: Maak Voorstelle-lêer oop en toe binne die lus.
 While om uit lêer te lees saam met begin en end = 1 punt, geen punte as daar
 geen begin en end is nie
 Aanvaar enige deel van teks wat na die Voorstelle-lêer geskryf word.
 Aanvaar die gebruik van die hele word vir toetsdoeleindes.

```
//=====
```

//VRAAG 3.4 (8)

```

procedure TfrmVraag3.btnBClick(Sender: TObject);
var
  sSoek   : String;
  k       : integer;
  bGevind : boolean;
begin
  sSoek := InputBox('Soek-navraag', 'Tik die
                    rekeningnommer in', '');
  redAfvoer.Lines.Clear;
  bGevind := false; ✓
  if NOT(ValideerRekNom(sSoek)) then✓
    ShowMessage('Ongeldige rekeningnommer') ✓
  else
    begin
      for k := 1 to iAantVerw do✓
        begin
          if (Pos(sSoek, arrVerw[k]) > 0) ✓ then
            begin
              redAfvoer.Lines.Add(arrVerw[k] + #9 +
                                arrInhoud[k]); ✓
              bGevind := true; ✓
            end;
          end;
        if bGevind = false then
          begin
            redAfvoer.Lines.Add('Geen kwessies is gerapporteer vir
                                rekeningnommer: ' + sSoek);
          end;
        end; // else
      end;
    end;

```

V 3.4

- (1) Inisialiseer Boolese veranderlike
- (1) Valideer rekeningnommer
- (1) Vertoon boodskap indien rekeningnommer ongeldig is
- (1) Binne lus
- (1) Toets of nommer wat ingetik is in skikking is en
- (1) Vertoon verw nommer en navraag
- (1) Verander Boolese waarde
- (1) Vertoon boodskap indien toevoerwaarde nie gevind is nie

Moet punt aftrek vir geen uppercase
 Aanvaar: Onttrek die rekeningnommer en vergelyk

```
end.
```

```
//=====
```

[40]

EINDE VAN AFDELING A: DELPHI

TOTAAL AFDELING A:

120

AFDELING B: JAVA**VRAAG 1: PROGRAMMERING EN DATABASIS**

```

import java.io.*;
import java.sql.*;
import javax.swing.*;
import java.util.Scanner;

public class ToetsVraag1
{
    public static void main (String[] args) throws SQLException,IOException
    {
        BufferedReader inKb = new BufferedReader (new InputStreamReader
                                                    (System.in));

        Damme DB = new Damme();
        System.out.println();
        char keuse = ' ';
        do
        {
            System.out.println("      KEUSELYS");
            System.out.println();
            System.out.println("      Opsie A");
            System.out.println("      Opsie B");
            System.out.println("      Opsie C");
            System.out.println("      Opsie D");
            System.out.println("      Opsie E");
            System.out.println("      Opsie F");
            System.out.println("      Opsie G");
            System.out.println();
            System.out.println("      V - VERLAAT");
            System.out.println(" ");
            System.out.print("      Jou Keuse? ");
            keuse = inKb.readLine().toUpperCase().charAt(0);
            System.out.println(" ");
            String sql = "";
            switch(keuse)
            {
                case 'A':
                    // VRAAG 1.1
                    {
                        sql ="SELECT *✓ FROM tblDamme✓ ORDER BY HoogteVanWal✓ ASC ";
                        DB.query(sql);
                        break;
                    }
                    //=====
                    (3)
                case 'B':
                    // VRAAG 1.2
                    {
                        System.out.print("Tik die naam van die provinsie in : ");
                        String pr = inKb.readLine();✓
                        sql = "SELECT DorpNaam, Bevolking✓ FROM tblDorpe WHERE✓
                            Bevolking > 100000✓ AND✓ Provinsie = '" + pr + "'"✓;
                        DB.query(sql);
                        break;
                    }
                    (6)
                    //=====
                case 'C':
                    // VRAAG 1.3
                    {
                        sql = "SELECT DamID, DamNaam✓, (YEAR(NOW()))✓ -
                            JaarVoltooi✓) AS Ouderdom✓, ROUND (DamVlak / Kapasiteit
                            * 100✓, 1✓) AS Persentasie✓ FROM tblDamme ";
                    }
                }
            }
        }
    }
}

```

Sien BYLAE A vir alternatiewe ennasienriglyne

```

        DB.query(sql);
        break;
    }
//=====
    case 'D':
        // VRAAG 1.4
        {
            sql = "SELECT Provinsie✓, COUNT(*)✓ AS KritiekeDorpe✓ FROM
                tblDorpe WHERE WaterBeperkings = TRUE✓ GROUP BY
                    Provinsie✓";

            DB.query(sql);
            break;
        }
//=====
    case 'E':
        // VRAAG 1.5
        {
            sql = "SELECT DISTINCT Provinsie✓ FROM tblDorpe✓, tblDamme✓
                WHERE tblDorpe.DamID✓ = tblDamme.DamID✓ AND Rivier✓
                    = 'Vaalrivier'✓";

            DB.query(sql);
            break;
        }
//=====
    case 'F':
        // VRAAG 1.6
        {
            sql = "UPDATE tblDorpe✓ SET✓ WaterBeperkings = True✓
                WHERE P rovinsie = 'Noordwes'✓";

            DB.query(sql);
            break;
        }
//=====
    case 'G':
        // VRAAG 1.7
        {
            sql = " DELETE✓ FROM tblDamme✓ WHERE HoogteVanWal <
                11.50✓";

            DB.query(sql);
            break;
        }
//=====
    }
    }while (keuse != 'V');

    DB.disconnect();
    System.out.println("Klaar");
}
}
//=====

```

[35]

VRAAG 2: OBJEK-GEORIËNTEERDE PROGRAMMERING**HuishoudingXXXX.java**

```
public class HuishoudingXXXX
{
    private String rekening;
    private int lede;
    private int [] arrWaterG;

    public HuishoudingXXXX()
    {
    }
}
```

// V 2.1.1**(3)**

```
public HuishoudingXXXX(String Rekening, int Lede, int [] arrWater)
{
    rekening = Rekening; ✓
    lede = Lede; ✓
    arrWaterG = arrWater; ✓
}
```

V 2.1.1

(3) Ken parameterwaardes toe aan privaat velde

Aanvaar 'n lus om aan skikking toe te ken
Trek net een punt af as die toekenningstellings verkeerd-om is bv.
arrWater := arrWaterG

//=====

// V 2.1.2**(4)**

Ignoreer enige foute in die definisie(verklaring) van metode - tel geen punte
Terugstuurtipe kan double of int wees

```
public int berekenTotaal()
{
    int totaal = 0; ✓
    for (int k = 0 ; k < arrWaterG.length; k++) ✓
    {
        totaal = totaal + arrWaterG[k]; ✓
        // of totaal += arrWaterG[k];
    }
    return totaal; ✓
}
```

V 2.1.2

(1) Inisialiseer totaal
(1) for lus
(1) Voeg element van skikking by totaal
(1) Stuur totaal terug

Aanvaar: totaal as 'n "instance" / globale veranderlike.
Aanvaar: lus tot <=7 of < 8
Aanvaar: Optel van skikkingwaardes individueel - geen lus
Aanvaar: gebruik nie lokale totaal veranderlike - Tel op en stuur terug -
alles in een stelling

Ken 4 punte toe as metode/kode reg is maar in die toetsklas is

//=====

// V 2.1.3**(2)**

```
public double ✓ berekenGem()
{
    return berekenTotaal() / 7.0; ✓
}
```

V 2.1.3

(1) Datatipe van terugstuurwaarde is
double /int
(1) Regte berekening

Aanvaar die gebruik van totaal slegs as berekenTotaal() geroep is (kan in die toetsklas geroep word).
 Aanvaar as die totaal weer hier bereken word.
 Aanvaar int as 'n terugstuurtipe - aanvaar /7 in plaas van /7.0

Ken 2 punte toe as metode/kode reg is maar in die toetsklas is

//=====

// V 2.1.4 (8/4=4) (na bo afgerond)

```
public int bepaalHoogsteDag() ✓
{
    int hoogDag = 1; ✓
    int hoogste = arrWaterG[0]; ✓
    for (int k=1; k < 7 ;k++) ✓
    {
        if (arrWaterG[k] > hoogste) ✓
        {
            hoogDag = k; ✓
            hoogste = arrWaterG[k]; ✓
        }
    }
    return hoogDag; ✓
}
```

V 2.1.4

- (1) Stuur int terug
- (1) Inisialiseer hoogDag
- (1) Inisialiseer hoogBedrag
- (1) for lus
- (1) if-stelling
- (1) Verander hoogDag
- (1) Verander hoogBedrag
- (1) Stuur hoogDag terug

Aanvaar sortering van hoeveelhede, as regte dag ook teruggestuur word - vol punte
 Aanvaar variasies wat reg is om die hoogste te kry bv. begin met 0 as hoogste ipv die eerste element.
 Sortering reg gedoen maar nie die regte dag gekry nie - 3 uit 4 punte

Ken 4 punte toe as metode/kode reg is maar in die toetsklas is

//=====

// V 2.1.5 (9)

```
public boolean bepaalHoogRisiko(double dagLimiet)
{
    double gem = berekenGem();
    int tel = 0; ✓
    for (int k = 0; k < arrWaterG.length;k++) ✓
    {
        if(arrWaterG[k] > dagLimiet) ✓
        {
            tel++; ✓
        }
    }
    if (gem > dagLimiet ✓ || ✓tel > 2) ✓
    {
        return true; ✓
    }
    else
    {
        return false; ✓
    }
}
```

V 2.1.5

- (1) Inisialiseer tel
- (1) Lus
- (1) if skikkingselement > dagLimiet
- (1) Inkrementeer tel
- (3) if gem > dagLimiet en tel > 2
- (1) Stuur true terug
- (1) else stuur false terug

Aanvaar globale veranderlikes
 Moenie punte aftrek vir toevoer van dagLimiet
 Aanvaar: if (berekenGem() > dagLimiet || tel > 2)
 Aanvaar: 'n Enkele stelling wat 'n Boolean waarde terugstuur
 return ✓ (gem > dagLimiet ✓ || ✓tel > 2) ✓✓
 Aanvaar: Initialiseer 'n Boolean veranderlike, stuur Boolean veranderlike terug

//=====

// V 2.1.6

(6)

1 punt vir elke stukkies inligting = 5 punte
1 punt om al die inligting saam te voeg in een string.

```
public String toString()
{
    String objStr = "Rekeningnommer: " +
                    reknom + "\nGetal lede: " + lede + "\n";
    objStr = objStr + "Daaglikse waterverbruik" + "\n"✓ + "Dae: " + "\t";
    for (int k = 1 ; k <= 7;k++)
    {
        objStr = objStr + k ✓ + "\t";
    }
    objStr = objStr + "\n" + "Waterverbruik:" ✓ + "\t";
    for (int k =0 ; k < arrWaterG.length;k++)✓
    {
        objStr = objStr + (arrWaterG[k] ✓ + "\t");
    }
    // Voeg stringe saam✓
    return objStr;
}
}
```

V 2.1.6

- (1) Opskrif + nuwe reël
- (1) Dagnommers
- (1) Opskrif
- (2) Waardes van skikking
- (1) Las stringe aanmekaar

Aanvaar die regte gebruik van die 'formatter' om die string saam te stel
Aanvaar aparte elemente van skikking in plaas van die lus.
Aanvaar enige korrekte vorm van saamvoeg van al die inligting in 'n string

```
//=====
```

ToetsVraag2XXXX.java

```
import java.util.Scanner;
```

```
public class ToetsVraag2XXXX
```

```
//=====
```

// V 2.2.1

(2)

```
public static void main(String args[]) throws Exception
{
```

```
    String reknom = "AC-23245";
```

```
    int lede = 4;
```

```
    int [] arrWaterG = {481, 438, 454, 353, 421, 396, 432};
```

```
    HuishoudingXXXX huishouding✓ = new HuishoudingXXXX(reknom, lede,
                                                         arrWaterG); ✓
```

Trek 1 punt af indien geen parameters.

```
    Scanner toevoer = new Scanner(System.in);
```

```
    char keuse = ' ';
```

```
    while (keuse != 'V')
```

```
    {
```

```
        System.out.println();
```

```
        System.out.println("        Keuselys");
```

```
        System.out.println(" ");
```

```
        System.out.println("        Opsie A ");
```

```
        System.out.println("        Opsie B ");
```

```
        System.out.println("        Opsie C ");
```

```
        System.out.println(" ");
```

```
        System.out.println("        V - VERLAAT");
```

V 2.2.1

- (2) Verklaar die objekveranderlike


```

System.out.println(" ");
System.out.print("      Jou keuse? ");

keuse = toevoer.next().toUpperCase().charAt(0);

```

```

switch (keuse)
{

```

// V 2.2.2

(4)

```

case 'A':
{
    System.out.println();
    System.out.println(huishouding.toString());✓
    System.out.println(" ");

    System.out.println("Totale waterverbruik: " ✓+
        huishouding.berekenTotaal()✓ + " liter");
    System.out.printf("Gemiddelde waterverbruik:%6.1f%s",
        huishouding.berekenGem()✓ , " liter");

    break;
}

```

V 2.2.2

- (1) Roep die toString metode van die objek
- (1) Vertoon die byskrif
- (1) Roep die berekenTotaal metode
- (1) Roep die berekenGem-metode

Aanvaar:Roep toString-metode as volg: System.out.println(huishouding)
Moenie streng wees met die bewoording van byskrifte of formattering van waardes nie

// V 2.2.3

(6)

```

case 'B':
{
    System.out.println();
    double gem = huishouding.berekenGem();✓
    System.out.println("Dae en hoeveelheid water
        waarmee die gemiddeld oorskry word");

    System.out.println("=====");
    System.out.printf("Gemiddelde waterverbruik per dag:%6.1f%s",
        huishouding.berekenGem()✓ , " liter");

    System.out.println("");
    System.out.println("Dae      Waarde oorskry gemiddeld met (liter)");

    for (int k = 0 ; k < arrWaterG.length;k++)✓
    {
        if (arrWaterG[k] > gem) ✓
        {
            System.out.printf("%-10d%-6.1f\n", (k+1), ✓
                (arrWaterG[k]- gem));✓
        }
    }
    System.out.println(" ");
    break;
}

```

V 2.2.3

- (1) Roep berekenGem()-metode
- (1) Vertoon gemiddeld
- (1) Lus
- (1) if
- (2) Vertoon getal & verskil

Geen punte vir opskrifte nie
Vertoon gemiddeld – Maak nie saak hoe die gemiddeld verkry is nie, punt is nie vir formattering nie
Vierde punt is vir berekening, nie vir formattering nie

// V 2.2.4

(5)

```

case 'C':
{
    System.out.println("Sleutel die limiet water per dag in ");
    double dagLimiet = toevoer.nextDouble();✓
    System.out.println( huishouding.toString());✓
    System.out.println(" ");

    System.out.println("Die dag waarop die
                        meeste water verbruik is: " +
                        huishouding.bepaalHoogsteDag());✓

    if ( huishouding.bepaalHoogRisiko(dagLimiet)) ✓
        System.out.println("Hoërisiko-huishouding");
    else
        System.out.println("Nie 'n hoërisiko-huishouding nie"); } ✓
    break;
}

```

V 2.2.4

- (1) Tik daglimiet in
- (1) roep toString
- (1) roep bepaalHoogsteDag()
- (1) If stelling
- (1) Vertoon regte boodskap

dagLimiet – int of double
 Tweede punt: Vir roep van toString – geen ander manier aanvaar om te vertoon
 Derde punt gaan vir roep van metode, nie byskrif. Aanvaar geen byskrif
 Vierde punt: vir roep van metode as deel van 'n if of toekenning aan veranderlike
 Vyfde punt: Vertoon van boodskap – punt vir twee boodskappe met else of tweede if

```

case 'V':
{
    System.exit(0);
} // case
} // switch
} // while
} // main
} // class

```

//=====

[45]

VRAAG 3: JAVA-PROGRAMMERING

LET WEL: Hierdie is slegs 'n voorbeeld – leerders kan hierdie vraag beantwoord op enige manier wat hulle wil. Gebruik die algemene rubriek in die nasienblaaie vir nasien.

ToetsVraag3XXXX.java**//VRAAG 3.1**

```
import java.io.*;
import java.util.*;

public class ToetsVraag3XXXX
{
    public void skepVoorstelLeer()
    {
        try
        {
            PrintWriter out = new PrintWriter(new FileWriter("Voorstelle.txt"));
        }
        catch(IOException e)
        {
            System.out.println("Fout met Voorstelle-Leer!!!" + e.getMessage());
        }
    }
}

/=====
```

//VRAAG 3.2**(6)**

```
public boolean valideerRekNom(String rekNom) ✓
{
    boolean geldigeRekNom = false; ✓
    if (rekNom.length() == 7 ✓ && Character.isLetter(rekNom.charAt(0))) ✓
    {
        geldigeRekNom = true; ✓
    }
    return geldigeRekNom; ✓
}
```

V 3.2

- (1) Opskrif van metode
- (1) Inisialiseer Boolese waarde
- (2) if stelling
- (1) Verander Boolese waarde
- (1) Stuur Boolese waarde terug

Aanvaar: if ... else in plaas van initialisering van Boolese waarde
 Aanvaar: Enige regte kode om eerste karakter te kry
 Aanvaar: Een stelling in metode wat die Boolese waarde terugstuur bv.
 return (rekNom.length() == 7 &&);

//=====

//VRAAG 3.3 **(24) + 2**

```
String [] arrVerw = new String [100];
String [] arrNavrae = new String [100];

int aantVerw = 0;
int telKlagtes = 0;
int telRekeninge = 0;

public void skepVerwysingsnommers()
{
    skepVoorstelLeer(); ✓
    try
    {
```

```

Scanner sc = new Scanner (new FileReader✓ ("DataA.txt"));✓
while (sc.hasNext())✓
{

String reel = sc.nextLine();✓
int dubbelpunt1 = reel.indexOf(":");✓
int dubbelpunt2 = reel.lastIndexOf(":");✓
String rekNom = reel.substring(dubbelpunt1+1,dubbelpunt2); ✓
int hash = reel.indexOf("#");
String datum = reel.substring(dubbelpunt2+1,hash); ✓
String navraag = reel.substring(hash+1); ✓

char tipe = reel.charAt(0); ✓
if (valideerRekNom(rekNom)) ✓
{
    if (tipe == 'V') ✓
    {
        try
        {
            PrintWriter out = new PrintWriter(new
                FileWriter("Voorstelle.txt",true)); ✓
            out.println(reel.substring(dubbelpunt1+1));✓
            out.close();✓
        }
        catch(IOException e)
        {
            System.out.println("Fout met Voorstelle-
                Leer!!!" + e.getMessage());
        }
    } //if
    else
    {
        tipe = Character.toUpperCase(tipe);
        switch(tipe) ✓
        {
            case 'K':telKlagtes++;✓
                arrVerw[aantVerw]✓= "K" +
                    telKlagtes + "-" + rekNom + "-" + datum; ✓

                break;

            case 'R': telRekeninge++;✓
                arrVerw[aantVerw] = "R" ✓+ telRekeninge + "-" +
                    rekNom + "-" + datum;

                break;

        }
        arrNavrae[aantVerw] = navraag; ✓
        aantVerw++;✓
    } //else
} //if
} //while
} //try
catch(FileNotFoundException e)
{
    System.out.println("Fout!!!" + e.getMessage());
}
System.out.println("\nVerwysingsnommers\n=====");
for (int i = 0; i<aantVerw;i++)✓
{
    System.out.println(arrVerw[i]); ✓
} //for
}

```

V 3.3

- (1) Skep Voorstelle-lêer
- (2) Maak Datalêer oop
- (1) While not eof
- (1) Lees 'n reël
- (1) Onttrek tipe kwessie
- (1) Onttrek rekeningnommer
- (1) Onttrek datum
- (1) Onttrek kwessie
- (1) Roep valideerReknom
- (1) Toets vir voorstel
- (1) Maak lêer oop vir skryf
- (1) Skryf voorstel na lêer
- (1) Maak lêer toe
- (1) Binne else vermeerder verwysingsnommer se teller
- (1) Kry eerste letter van kwessie
- (1) Toets kategorie
- (2) Skep verwysingsnommer vir klagte
- (2) Skep verwysingsnommer vir rekening
- (1) Skep kwessie verwysingsnommer
- (1) Stoor verwysingsnommer in skikking
- (1) Stoor navraag in skikking
- (2) Vertoon

Aanvaar: Maak Voorstelle-lêer oop en toe binne die lus.
 While om uit lêer te lees saam met begin en end = 1 punt, geen punte as daar geen begin en end is nie
 Aanvaar enige deel van teks wat na die Voorstelle-lêer geskryf word.
 Aanvaar die gebruik van die hele word vir toetsdoeleindes.

//=====

//VRAAG 3.4

(8)

```
public void seekRekening()
{
    Scanner kb = new Scanner (System.in);
    System.out.println("\nTik die rekeningnommer in ");
    String rekNom = kb.next();
    boolean gevind = false; ✓
    System.out.println();

    if !(valideerRekNom(rekNom)) ✓
        System.out.println("Ongeldige rekeningnommer"); ✓

    else
    {
        for (int i = 0; i< aantVerw; i++)✓
        {
            if (arrVerw[i].contains(rekNom)) ✓
            {
                System.out.println(arrVerw[i]+"\\t"+
                                   arrNavrae[i]); ✓
                gevind =true; ✓
            }
        }
        ///for

        if (!gevind)
        {
            System.out.println("Geen kwessie is gerapporteer vir
                               rekeningnommer:" + rekNom);
        }
    } //else
}
```

V 3.4

- (1) Inisialiseer Boolse veranderlike
- (1) Valideer rekeningnommer
- (1) Vertoon boodskap indien rekeningnommer ongeldig is
- (1) Binne lus
- (1) Toets of nommer ingetik in skikking is en
- (1) Vertoon verw nommer en navraag
- (1) Verander Boolse waarde
- (1) Vertoon boodskap indien toevoerwaarde nie gevind is nie

} ✓

Aanvaar: if(arrVerw[i].indexOf(rekNom)>-1)
 Aanvaar: Onttrek die rekeningnommer en vergelyk

//=====

```
public static void main (String [] args)
{
    ToetsVraag3XXXX obj = new ToetsVraag3XXXX();
    Scanner toevoer = new Scanner(System.in);
    boolean opsieA = false;
    char keuse = ' ';
    while (keuse != 'V')
    {
        System.out.println();
        System.out.println("                Keuselys");
        System.out.println(" ");
        System.out.println("                Opsie A");
        System.out.println("                Opsie B");
        System.out.println(" ");
        System.out.println("                V - Verlaat");
        System.out.println(" ");
    }
```

```
System.out.print("      Jou keuse? ");
keuse = toevvoer.nextLine().toUpperCase().charAt(0);

if (keuse == 'A')
{
    obj.skepVerwysingsnommers();
    opsieA = true;
}

if(keuse == 'B')
{
    if (!(opsieA))
    {
        System.out.println("\n\nEerste keuse Opsie A ");
    }
    else
    {
        obj.soekRekening();
    }
}

if( keuse == 'V')
{
    System.exit(0);
}
} // while
} // class
//=====
```

[40]**EINDE VAN AFDELING B: JAVA****TOTAAL AFDELING B: 120**
GROOTTOTAAL: 120

BYLAE A

VRAAG 1: DELPHI – PROGRAMMERING EN DATABASIS

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG 1: DELPHI – NASIENBLAD			
In die algemeen: Trek slegs 1 punt af vir 'n algemene fout wat regdeur in al die SQL-stellings gemaak word. Trek 'n maksimum van een punt af indien geen punt in die memo toegeken is nie maar 'n fout is gemaak,			
VRAAG	ASPEK	MAKS. PUNTE	LEERDER SE PUNTE
1.1	SELECT *✓ FROM tblDamme✓ ORDER BY HoogteVanWal ASC✓	3	
1.2	Toevoer van provinsie✓ SELECT DorpNaam, Bevolking✓ FROM tblDorpe WHERE✓ Bevolking > 100000✓ AND✓ Provinsie = '' + pr + ''✓ Aanvaar: Provinsie LIKE Laaste punt: Aanvaar string in aanhalings "100000" is verkeerd, geen aanhalings Volgorde van geselekteerde velde nie belangrik nie	6	
1.3	'SELECT DamID, DamNaam, ✓ (YEAR(NOW()))✓ - JaarVoltooi✓) AS Ouderdom✓, ROUND(DamVlak / Kapasiteit * 100✓, 1✓) AS Persentasie✓ FROM tblDamme Nota: SELECT DamID,DamNaam ...FROM tblDamme (een konsep, 1 punt) Nuwe veldname(alle vrae)-moenie penaliseer indien nie presies soos voorgestel in vraag. Aanvaar: YEAR(DATE())of 2011 Aanvaar: format (DamVlak / Kapasiteit * 100, "0.0") Aanvaar: Regte gebruik van int on af te rond na 1 desimaal: Int((DamVlak / Kapasiteit * 100)*10)/10	7	
1.4	SELECT Provinsie✓, COUNT(*)✓ AS KritiekeDorp✓ FROM tblDorpe WHERE WaterBeperkings = TRUE ✓GROUP BY Provinsie✓ Aanvaar: WaterBeperkings = YES of NO Aanvaar: COUNT(enige veld uit tabel ipv *) Aanvaar: WHERE WaterBeperkings (sonder = True) Group By moet aan die einde wees	5	
1.5	SELECT DISTINCT Provinsie✓ FROM tblDorpe✓, tblDamme✓ WHERE tblDorpe.DamID✓ = tblDamme.DamID✓ AND Rivier✓ = "Vaalrivier"✓ Aanvaar: GROUP BY Provinsie aan einde van stelling ipv DISTINCT Provinsie Aanvaar: INNER JOIN om tabelle te koppel ...FROM tblDamme INNER JOIN tblDorpe ON tblDamme.DamID = tblDorpe.DamID.... Aanvaar: LIKE "Vaal%" Trek een punt af vir sintaksfout, bv. Gebruik '_' ipv '.' Of uitlaat van '.', ens Aanvaar indien aliasse gebruik word, bv. tblDorpe A, tblDamme B	7	

1.6	UPDATE tblDorpe✓ SET✓ WaterBeperkings = True✓ WHERE Provinsie = "Noordwes" ✓ Aanvaar: Provinsie LIKE "Noordwes" Aanvaar: WaterBeperkings = YES of NO Noordwes moet reg gespel wees, in aanhalings	4	
1.7	DELETE✓ FROM tblDamme✓ WHERE HoogteVanWal < 11.50✓ Aanvaar: Delete *	3	
	TOTAAL:	35	

BYLAE B

VRAAG 2: DELPHI – OBJEK-GEORIËNTEERDE PROGRAMMERING

(Sien na tesame met die opmerkings in die proef antwoord op bladsye 4 - 8)

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG 2: DELPHI – NASIENBLAD			
VRAAG	ASPEK	MAKS. PUNTE	LEERDER SE PUNTE
2.1			
2.1.1	Konstruktor: (3) Ken parameters toe aan privaat velde	3	
2.1.2	berekenTotaal: (1) Inisialiseer totaal (1) for lus (1) Voeg skikkingselement by totaal (1) stuur totaal terug	4	
2.1.3	berekenGem: (1) Datatipe van die terugstuurwaarde is real (of double) (1) Regte berekening	2	
2.1.4	bepaalHoogsteDag: (1) Stuur heelgetal (integer) terug (1) Inisialiseer iHoogsteDag (1) Inisialiseer iHoogste (1) For lus (1) if-stelling (1) Verander iHoogsteDag (1) Verander iHoogste (1) Stuur iHoogsteDag terug	8/2=4 (rond af na bo)	
2.1.5	bepaalHoogRisiko: (1) Inisialiseer teller (1) Lus (1) if skikkingselement > dagLimiet (1) Inkrementeer teller (3) if gem > dagLimiet or teller > 2 (1) Stuur true terug (1) else Stuur false terug	9	
2.1.6	toString: (1) Opskrif + nuwe reel (1) Dagnommers (1) Opskrif (2) Waardes van skikking (1) Las stringe aanmekaar	6	
2.2			
2.2.1	(2) Verklaar die objekveranderlike	2	
2.2.2	(1) Roep die toString metode van die objek (1) Vertoon die byskrif (1) Roep die berekenTotaal metode (1) Roep die berekenGem-metode	4	
2.2.3	(1) Roep berekenGem-metode (1) Vertoon gemiddeld (1) Lus (1) if (2) Vertoon getal & verskil	6	
2.2.4	(1) Tik daglimiet in (1) roep toString (1) roep bepaalHoogsteDag((1) If stelling (1) Vertoon regte boodskap	5	
	TOTAAL:	45	

BYLAE C**VRAAG 3: DELPHI-PROGRAMMERING****(Sien na tesame met die opmerkings in die proef antwoord op bladsye 9 - 13)**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG 3: DELPHI – NASIENBLAD			
VRAAG	ASPEK	MAKS. PUNTE	LEERDER SE PUNTE
3.1	Kode is gegee in die Afrikaans Java weergawe 2 punte is hertoegeken in Vraag 3.3		
3.2	(1) Opskrif van subprogram (1) Inisialiseer Boolese waarde (2) if stelling (1) Verander Boolese waarde (1) Stuur Boolese waarde terug	6	
3.3	Opsie A: (1) Roep metode om Voorstelle-lêer te skep (2) Maak Data-lêer oop (1) Maak Voorstelle-lêer oop (1) While not eof (1) Lees 'n reël (1) Onttrek tipe kwessie (1) Onttrek rekeningnommer (1) Onttrek datum (1) Onttrek kwessie (1) Roep valideerReknom (1) Toets vir voorstel (1) Skryf voorstel na lêer (1) Binne else vermeerder verwnommer se teller (1) Kry eerste letter van kwessie (1) Toets kategorie (2) Skep verwysingsnommer vir klagte (2) Skep verwysingsnommer vir rekening (1) Skep kwessie verwysingsnommer (1) Stoor verwysingsnommer in skikking (1) Stoor navraag in skikking (2) Vertoon verwysingsnommers (1) Maak Voorstelle-lêer toe	24 + 2	
3.4	Opsie B: (1) Inisialiseer Boolese veranderlike (1) Valideer reknommer (1) Vertoon boodskap as reknommer ongeldig is (1) Binne lus (1) Toets of nommer ingetik in skikking is en (1) vertoon verw nommer en navraag (1) Verander Boolese waarde (1) Vertoon boodskap indien toevoerwaarde nie gevind is nie	8	
	TOTAAL:	40	

BYLAE D

VRAAG 1: JAVA – PROGRAMMERING EN DATABASIS

SENTRUMNOMMER:		EKSAMENNUMMER:	
VRAAG 1: JAVA – NASIENBLAD			
In die algemeen: Trek slegs 1 punt af vir 'n algemene fout wat regdeur in al die SQL-stellings gemaak word. Trek 'n maksimum van een punt af indien geen punt in die memo toegeken is nie maar 'n fout is gemaak,			
VRAAG	ASPEK	MAKS. PUNTE	LEERDER SE PUNTE
1.1	SELECT *✓ FROM tblDamme✓ ORDER BY HoogteVanWal✓ ASC	3	
1.2	Toevoer Provinsie ✓ SELECT DorpNaam, Bevolking✓ FROM tblDorpe WHERE✓ Bevolking > 100000✓ AND✓ Provinsie = '"+ pr + "'✓; Aanvaar:Provinsie LIKE Laaste punt: Aanvaar string in aanhalings "100000" is verkeerd, geen aanhalings Volgorde van geselekteerde velde nie belangrik nie	6	
1.3	SELECT DamID, DamNaam✓, (YEAR(NOW()))✓ - JaarVoltooi✓) AS Ouderdom✓, ROUND (DamVlak / Kapasiteit * 100✓, 1✓) AS Persentasie✓ FROM tblDamme Nota: SELECT DamID,DamNaam ...FROM tblDamme (een konsep, 1 punt) Nuwe veldname(alle vrae)-moenie penaliseer indien nie presies soos voorgestel in vraag. Aanvaar: YEAR(DATE())of 2011 of YEAR(NOW) Aanvaar: FORMAT (DamVlak / Kapasiteit * 100, "0.0") Aanvaar: Regte gebruik van int on af te rond na 1 desimaal: Int((DamVlak / Kapasiteit * 100)*10)/10	7	
1.4	SELECT Provinsie✓, COUNT(*)✓ AS KritiekeDorpe✓ FROM tblDorpe WHERE WaterBeperkings = TRUE✓ GROUP BY Provinsie✓ Aanvaar: WaterBeperkings = YES of NO Aanvaar: COUNT(enige veld uit tabel ipv *) Aanvaar: WHERE WaterBeperkings (sonder = True) GROUP BY moet aan die einde wees	5	
1.5	SELECT DISTINCT Provinsie✓ FROM tblDorpe✓, tblDamme✓ WHERE tblDorpe.DamID✓ = tblDamme.DamID✓ AND Rivier✓ = 'Vaalrivier'✓ Aanvaar: GROUP BY Provinsie aan einde van stelling ipv DISTINCT Provinsie Aanvaar: INNER JOIN om tabelle te koppel ...FROM tblDamme INNER JOIN tblDorpe ON tblDamme.DamID = tblDorpe.DamID.... Aanvaar: LIKE "Vaal%" Trek een punt af vir sintaksfout, bv. Gebruik `_' ipv `.' Of uitlaat van `.', ens Aanvaar indien aliasse gebruik word, bv. tblDorpe A, tblDamme B	7	

1.6	UPDATE tblDorpe✓ SET✓ WaterBeperkings = True✓ WHERE Provinsie = 'Noordwes'✓ Aanvaar: Provinsie LIKE 'Noordwes' Aanvaar: WaterBeperkings = YES of NO Noordwes moet reg gespel wees en in aanhalings	4	
1.7	DELETE✓ FROM tblDamme✓ WHERE HoogteVanWal < 11.50✓"; Aanvaar 999999DELETE *	3	
	TOTAAL:	35	

BYLAE E

VRAAG 2: JAVAOBJEK-GEORIËNTEDE PROGRAMMERING

(Sien na tesame met die opmerkings in die proef antwoord op bladsye 14 - 18)

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG 2: JAVA – NASIENBLAD			
VRAAG	ASPEK	MAKS. PUNTE	LEERDER SE PUNTE
2.1			
2.1.1	Konstruktor: (3) Ken parameters toe aan privaat velde	3	
2.1.2	berekenTotaal: (1) Inisialiseer totaal (1) for lus (1) Voeg skikkingselement by totaal (1) stuur totaal terug (gebruik result of die naam van die funksie)	4	
2.1.3	berekenGem: (1) Datatipe van die terugstuurwaarde is real (of double) (1) Regte berekening	2	
2.1.4	bepaalHoogsteDag: (1) Stuur heelgetal (integer) terug (1) Inisialiseer iHoogsteDag (1) Inisialiseer iHoogste (1) For lus (1) if-stelling (1) Verander iHoogsteDag (1) Verander iHoogste (1) Stuur iHoogsteDag terug	8/2=4 (rond af na bo)	
2.1.5	bepaalHoogRisiko: (1) Inisialiseer teller (1) Lus (1) if skikkingselement > dagLimiet (1) Inkrementeer teller (3) if gem > dagLimiet or teller > 2 (1) Result true (1) else Result false	9	
2.1.6	toString: (1) Opskrif + nuwe reel (1) Dagnommers (1) Opskrif (2) Waardes van skikking (1) Las stringe aanmekaar	6	
2.2			
2.2.1	(2) Verklaar die objekveranderlike	2	
2.2.2	(1) Roep die toString metode van die objek (1) Vertoon die byskrif (1) Roep die berekenTotaal metode (1) Roep die berekenGem-metode	4	
2.2.3	(1) roep berekenGem-metode (1) Vertoon gemiddeld (1) Lus (1) if (2) Vertoon getal & verskil	6	
2.2.4	(1) Tik daglimiet in (1) Roep toString (1) Roep bepaalHoogsteDag (1) If stelling (1) Vertoon regte boodskap	5	
	TOTAAL	45	

BYLAE F**VRAAG 3: JAVA – PROGRAMMERING****(Sien na tesame met die opmerkings in die proef antwoord op bladsye 19 - 22)**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG 3: JAVA – NASIENBLAD			
VRAAG	ASPEK	MAKS. PUNTE	LEERDER SE PUNTE
3.1	Kode is gegee in die Afrikaans Java weergawe 2 punte is hertoegeken in Vraag 3.3		
3.2	(1) Opskrif van metode (1) Inisialiseer Boolese waarde (2) if stelling (1) Verander Boolese waarde (1) Stuur Boolese waarde terug	6	
3.3	Opsie A: (1) Roep metode om Voorstelle-lêer te skep (2) Maak Data-lêer oop om uit te lees (1) While nog teks om te lees (1) Lees 'n reël (1) Onttrek tipe kwessie (1) Onttrek rekeningnommer (1) Onttrek datum (1) Onttrek kwessie (1) Roep valideerReknom (1) Toets vir voorstel (1) Maak lêer oop vir skryf (1) Skryf voorstel na lêer (1) Maak lêer toe (1) Binne else vermeerder verwnommer se teller (1) Kry eerste letter van kwessie (1) Toets kategorie (2) Skep verwysingsnommer vir klagte (2) Skep verwysingsnommer vir rekening (1) Skep kwessie verwysingsnommer (1) Stoor verwysingsnommer in skikking (1) Stoor navraag in skikking (2) Vertoon verwysingsnommers	24 + 2	
3.4	Opsie B: (1) Inisialiseer Boolese veranderlike (1) Valideer reknommer (1) Vertoon boodskap as reknommer ongeldig is (1) Binne lus (1) Toets of nommer ingetik in skikking is en (1) vertoon verw nommer en navraag (1) Verander Boolese waarde (1) Vertoon boodskap indien toevoerwaarde nie gevind is nie	8	
	TOTAAL:	40	