GAUTENGSE DEPARTEMENT VAN ONDERWYS
VOORBEREIDENDE EKSAMEN

INLIGTINGSTEGNOLOGIE
(Eerste Vraestel: Prakties)

MOONTLIKE ANTWOORDE

ALGEMENE INLIGTING

- Addenda A tot D bevat die dekblad en die **nasienrubrieke** ("*grids*") vir elke vraag. Kopieë moet vir elke kandidaat gemaak word om te voltooi tydens die nasiensessie.
- **ALLE WERKBARE oplossings moet noukeurig oorweeg word**. Daar mag alternatiewe oplossings wees vir dié wat in die gegewe memo voorsien word.
- **Sintaksfoute** moet slegs gepenaliseer word indien dit lei tot 'n logiese fout, tot 'n maksimum van 2 punte per vraag. Alle ander sintaksfoute moet geïgnoreer word.
- By Vraag 2 en 3 moet die halfpunte oorgedra word na die dekblad. ***SLEGS die finale totaal moet afgerond word op die dekblad***.

VRAAG 1: DATABASIS EN PROGRAMMERING

```
unit Q1_DUM_U;
{voorgestelde memorandum: Vraag 1}
interface
   //definisie verklaring van vorm en prosedures
implementation
{$R *.dfm}
procedure TQ1.Button1Click(Sender: TObject);
begin
//vraag 1.1
  adoQ.Close;
  adoQ.SQL.Text := 'SELECT Dokter, Tarief ' ✓ +
                   'FROM Dokters ' ✓+
                   'ORDER BY ✓ Dokter DESC' ✓;
  adoQ.Open;
end;                                                    (4)
//=========================================================
procedure TQ1.Button2Click(Sender: TObject);
var
  Jaar : String;
begin
//vraag 1.2
  Jaar := InputBox('Geboortes', 'Vir watter jaar', '2008'); ✓
  adoQ.Close;
  adoQ.SQL.Text := 'SELECT Van, Voornaam, Geb, Gewig '+
                   'FROM Babas ' +                    ✓
                   'WHERE (YEAR(Geb)✓ = ' + Jaar✓ + ') AND✓ (Gewig < 3.5)✓';
  adoQ.Open;
end;                                                    (6)
//=========================================================
procedure TQ1.Button3Click(Sender: TObject);
begin
//vraag 1.3
  adoQ.Close;
  adoQ.SQL.Text := 'SELECT Count(*)✓ AS [Seuns gebore in Jan 2008]✓ ' +
                   'FROM Babas ' +
                   'WHERE (Geslag = "M")✓ AND ' +
            '(Year(Geb) = 2008) AND (Month(Geb) = 1)'✓;
  adoQ.Open;
end;                                                (4)
//=========================================================
```

Let op:
Die bepaling van die datum (Jan 2008) kan baie alternatiewe oplossings hê:

- (Geb LIKE "2008/01/%")
- Geb in (DateValue("2008/01/01"), DateValue("2008/01/31")

TOETS ELKE ALTERNATIEF!!!

```
procedure TQ1.Button4Click(Sender: TObject);
begin
//vraag 1.4
  adoQ.Close;
  adoQ.SQL.Text := 'UPDATE Babas '✓ +
                   'SET ✓ Geb = #2007/06/16# ✓' +
                   'WHERE (Van = "Adam") AND (Voornaam = "Stanley")'✓;
  adoQ.ExecSQL;
  adoQ.Close;
  adoQ.SQL.Text := 'SELECT * FROM Babas ' +
                   'WHERE (Van = "Adam") AND (Voornaam = "Stanley")';✓{vertoon}
  adoQ.Open;
end;                                                    (5)
//=========================================================

procedure TQ1.Button5Click(Sender: TObject);
begin
//vraag 1.5
  adoQ.Close;
  adoQ.SQL.Text := 'SELECT Voornaam, Geb, Dokter,✓ ' +
           'Format✓(DaeInHosp * Tarief✓, "Currency") AS [Koste per kind]✓'+
                   'FROM Dokters, Babas '✓ +
                   'WHERE (Van = "Green")✓ AND (DokterID = PraktykNom)✓';
  adoQ.Open;
end;                   (7)
//==============================={FROM OUERS INNER JOIN BABAS ON OUERS.REKNOM = BABAS.REKNOM}

procedure TQ1.Button6Click(Sender: TObject);
begin
//vraag 1.6
  adoQ.Close;
  adoQ.SQL.Text := 'SELECT TOP 5 Babas.Van, Voornaam, Geb AS [Geboortedatum], ' +
                           'Woonbuurt ✓' +
                   'FROM Ouers, Babas ' +
                   'WHERE (Ouers.RekNom = Babas.RekNom) ✓AND ' +
                          '(Geslag = "F")✓ AND✓ ' +
              '((Woonbuurt = "Sinoville")✓ OR (Woonbuurt = "Rooihuiskraal")✓)'+
                    'ORDER BY Geb DESC' ✓;
                                 {FROM OUERS INNER JOIN BABAS ON OUERS.REKNOM = BABAS.REKNOM}
  adoQ.Open;
end;                                                    (7)
//=========================================================

procedure TQ1.Button7Click(Sender: TObject);
begin
//vraag 1.7
  adoQ.Close;
  adoQ.SQL.Text := 'INSERT INTO Dokters ✓ (PraktykNom, Dokter, Tarief)✓ ' +
                   'VALUES ("BEL763", "BELL G", 550.00)✓';
  adoQ.ExecSQL;
  adoQ.Close;
  adoQ.SQL.Text := 'Select * FROM Dokters Order BY Dokter'; ✓
  adoQ.Open;
end;                                                    (4)
//=========================================================

procedure TQ1.Button8Click(Sender: TObject);
begin
//vraag 1.8
  adoQ.Close;
  adoQ.SQL.Text := 'SELECT Van, Voornaam ' +✓
                   'FROM Babas ' +
                   'WHERE (Voornaam Like "Jo%"✓) AND (Geslag = "M")'✓;
  adoQ.Open;
end;                                                    (3)
//=========================================================
```

```
procedure TQ1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
//maak die konneksie met databasis toe
   adoQ.Close;
   adoCONN.Close;
end;

procedure TQ1.FormCreate(Sender: TObject);
begin
//Skep die connectionstring outomaties.
   IF FileExists(ExtractFilePath(Application.ExeName) + 'BABAINFO.mdb')
   then
    begin

         adoCONN.ConnectionString :=
                'Provider=Microsoft.Jet.OLEDB.4.0;Data Source=' +
                'BABAINFO.mdb;Persist Security Info=False';
         adoCONN.Open;
         stbQ1.SimpleText := 'Databasis is gekoppel.';
    end
   else
     begin
      stbQ1.SimpleText := 'Databasis NIE gekoppel NIE!';
     end;
end;

end.
```

Gegewe kode
Geen punte

VRAAG 2: OOP

CLASS-UNIT

```
unit DAGSORG_DUM_JH;
{voorgestelde memorandum: Vraag 2 - Class}
interface

TYPE
   TPasient = class(TObject) ✓
     private
       fVolleNaam : String[30]; ✓
       fFisio,
       fRadio,
       fSpesialis : Boolean; ✓
       fBalans    : Real; ✓
     public
       Constructor Create(Naam : String; Fisio, Radio, Spes : Boolean);
       function GetVolleNaam : String;
       function GetBalans    : Real;
   end;

implementation

{ TPasient }

constructor TPasient.Create(Naam: String✓; Fisio, Radio, Spes: Boolean✓);
var
    F, R, S : real;
begin
    fVolleNaam := Naam; ✓
    fFisio     := fisio;
    fRadio     := Radio;
    fSpesialis := Spes; ✓{al 3 boolean velde}
    fBalans    := 0;

   IF Fisio ✓
    then F := 325.00 ✓
    else F := 0; ✓
                    {✓ vir herhaling van toeken}
   IF Radio
    then R := 423.00
    else R := 0;
```

```
   IF Spes
    then S := 876.00
    else S := 0;

   fBalans := F + R + S; ✓✓
end;

function TPasient.GetBalans: Real; ✓
begin
    Result := fBalans; ✓
end;

function TPasient.GetVolleNaam: String;  ✓
begin
    Result := fVolleNaam; ✓
end;
end.
```

FORM-UNIT

```
unit VR2_DUM_U;
{voorgestelde memorandum: Vraag 2}
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, ComCtrls, ExtCtrls, StdCtrls, Buttons,
  DAGSORG_DUM_JH; ✓

type
  TVRAAG2 = class(TForm)
    pcVr2: TPageControl;
    tbsRegistrasie: TTabSheet;
    tbsOntslag: TTabSheet;
    MainMenu1: TMainMenu;
    Verlaatprogram1: TMenuItem;
    bmbRegistreer: TBitBtn;
    edtNaam: TEdit;
    Label1: TLabel;
    btnNaamlys: TButton;
    redNaamlys: TRichEdit;
    GroupBox1: TGroupBox;
    chkFisio: TCheckBox;
    chkRadio: TCheckBox;
    chkSpes: TCheckBox;
    lblAantal: TLabel;
    GroupBox2: TGroupBox;
    Label3: TLabel;
    edtNr: TEdit;
    bmbVerwyder: TBitBtn;
    rgpOntslaan: TRadioGroup;
    procedure bmbRegistreerClick(Sender: TObject);
    procedure Verlaatprogram1Click(Sender: TObject);
    procedure btnNaamlysClick(Sender: TObject);
    procedure bmbVerwyderClick(Sender: TObject);
    procedure edtNrKeyPress(Sender: TObject; var Key: Char);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }

    Naamlys : Array[1..50]✓ of TPasient✓;
    Aantal  : Integer;✓

    iOorgeplaas, iRegistreer,
    iOntslaan         : Integer;
    Totaal            : Real;
  public
    { Public declarations }
  end;

var
  VRAAG2: TVRAAG2;

implementation
```

```
{$R *.dfm}

procedure TVRAAG2.bmbRegistreerClick(Sender: TObject);
begin

   IF (Aantal > 50) ✓
   then
    begin
      ShowMessage('Die maksimum aantal pasiente van 50 is bespreek.');✓
      Exit;
    end;

   IF NOT chkFisio.Checked AND
      NOT chkRadio.Checked AND   ✓✓
      NOT chkSpes.Checked
   then
    begin
      ShowMessage('Moet vir tenminste EEN behandeling registreer!');✓
      Exit;
    end;

   IF Length(edtNaam.Text) = 0 ✓
   then
    begin
      ShowMessage('Die naam van die pasient is nie ingevul nie.'); ✓
      Exit;
    end;

   Inc(Aantal, 1);✓
   Inc(iRegistreer, 1);
   Naamlys[Aantal]✓ := TPasient.Create✓(edtNaam.Text, chkFisio.Checked,
                                        chkRadio.Checked,chkSpes.Checked)✓✓;
end;

procedure TVRAAG2.bmbVerwyderClick(Sender: TObject);
var
   iX,a     : Integer;
   slyn   : String;
   TF     : TextFile; ✓
begin
//verwyder pasient
   iX := StrToInt(edtNr.Text);
   IF NOT(ix in [1..Aantal])
    then
      begin                    ✓✓✓✓
        showMessage('Ongeldige toevoer: 1..' + IntToStr(Aantal));
      ✓ Exit;
      end;
  //geldige nommer
  ✓case rgpOntslaan.ItemIndex of
    0✓ : begin
          //skryf na teksfile
          AssignFile(TF, 'WaakEenheid.txt');✓
          IF FileExists('WaakEenheid.txt') ✓
           then Append(TF)✓
           ✓else Rewrite(TF)✓;
          sLyn := Naamlys[iX].GetVolleNaam + ' is oorgeplaas na waakeenheid.';✓
          Writeln✓(TF, Naamlys[iX].GetVolleNaam✓);

          CloseFile(TF);✓
          inc(iOorgeplaas)✓; (punt gebruik by 2.3)
        end; //na waakeenheid
    1✓ : begin
          sLyn := Naamlys[iX].GetVolleNaam + ' is ontslaan.';✓
          inc(iOntslaan); ✓ (punt gebruik by 2.3)
          Totaal := Totaal + Naamlys[iX].GetBalans;✓ (punt gebruik by 2.3)
        end; //ontslaan
  end;
  //verwyder uit skikking
  Naamlys[iX].Free; ✓
  For a := iX to Aantal do✓
    Naamlys[a] := Naamlys[a+1]; ✓✓
  Dec(Aantal, 1);  ✓
    MessageDlg(sLyn, mtInformation, [mbOk], 0);✓
```

```
  btnNaamlys.Click; ✓ //refresh display!
end;

procedure TVRAAG2.btnNaamlysClick(Sender: TObject);
var
  a : integer;
begin
   redNaamlys.Lines.Clear;✓
   redNaamlys.Paragraph.TabCount := 1;
   redNaamlys.Paragraph.Tab[0]   := 100;          } ✓{kolomme}
   redNaamlys.SelAttributes.Style := [fsBold];
   redNaamlys.Lines.Add('Pasient' + #9 + 'Balans'); ✓{opskrifte}
   for a := 1 to Aantal✓ do
     IF (Naamlys[A] is TPasient)
      then
       redNaamlys.Lines.Add✓(InttoStr(a)✓ + '. ' + Naamlys[a].GetVolleNaam ✓+ #9 +
                            'R' + Format✓('%8.2f'✓, [Naamlys[a].GetBalans])✓)
      else
       redNaamlys.Lines.Add(IntToStr(a) + 'Geen objeck');

   lblAantal.Caption := 'Aantal = ' + IntToStr(Aantal);✓
end;

procedure TVRAAG2.edtNrKeyPress(Sender: TObject; var Key: Char);
begin
   IF NOT(Key in ['0'..'9', #8]) } ✓✓✓✓      Alternatiewelik gebruik TRY...EXCEPT of
    then Key := #0;               }           VAL om gedige numeriese data te toets.
end;

procedure TVRAAG2.FormCreate(Sender: TObject);
begin
   iOorgeplaas := 0;
   iOntslaan   := 0;
   Totaal      := 0;✓   (punt gebruik by 2.3)
end;

procedure TVRAAG2.Verlaatprogram1Click(Sender: TObject);
var
   a : integer;
   sLyn        : String;
begin
  //display boodskap
  slyn := 'Datum: ' + FormatDateTime✓('dd mmm yyyy'✓, Date) + #13 + #13 + ✓{oop reel}
         ✓'Pasiente vir die dag = ' + Inttostr(iRegistreer) + #13 + #13 +
          'Aantal oorgeplaas na waakeenheid = ' + IntToStr(ioorgeplaas)✓ + #13 + #13 +
          'Aantal ontslaan = ' + InttoStr(iOntslaan) + #13 + #13 +
          'Totale inkomste = ' + FloattostrF(Totaal, ffCurrency✓, 8, 2✓);
  MessageDlg(slyn, mtInformation, [mbOk], 0);✓

  //verwyder alle objects
  for a := 1 to 50 do✓
   Naamlys[a].Free;✓

  //end program
  Application.Terminate;✓
end;

end.
```

VRAAG 3: DELPHI

```
unit VR3_DUM_U;
{voorgestelde memorandum: Vraag 3}
interface

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

type
  TVRAAG_3 = class(TForm)
    Panel1: TPanel;
```

```
    BitBtn1: TBitBtn;
    Panel2: TPanel;
    GroupBox1: TGroupBox;
    edtNommer: TEdit;
    bmbVerslag: TBitBtn;
    gpbSave: TGroupBox;
    bmbSaveAs: TBitBtn;
    bmbPrint: TBitBtn;
    dlgSaveAs: TSaveDialog;
    redVerslag: TRichEdit;
    dlgPrint: TPrintDialog;
    procedure FormCreate(Sender: TObject);
    procedure bmbVerslagClick(Sender: TObject);
    procedure bmbSaveAsClick(Sender: TObject);
    procedure bmbPrintClick(Sender: TObject);
  private
    { Private declarations }
    sPers_Info                    : String;
    function Geldige_Rekening(sNomX : String) : Boolean; {gegewe kode}
  public
    { Public declarations }
  end;

var
  VRAAG_3: TVRAAG_3;

implementation

{$R *.dfm}

procedure TVRAAG_3.bmbPrintClick(Sender: TObject);
begin
  //Maak 'n drukstuk van verslag van baba X?
  IF✓ dlgPrint.Execute✓
  then redVerslag.Print('Verslag');✓✓
end;

procedure TVRAAG_3.bmbSaveAsClick(Sender: TObject);
begin
  //Stoor die verslag van baba X?
  if✓ dlgSaveAs.Execute ✓
   then
      redVerslag.Lines.SaveToFile (dlgSaveAs.FileName);✓
    end;
end;

procedure TVRAAG_3.bmbVerslagClick(Sender: TObject);
var
    sLyn, sNom,
    sV, sN, sG, sB : String;
    TF             : TextFile;
    Tel, a, b          : Integer;
    arrTyd       : Array[1..100] of TDateTime;      ✓ {declaration van enige skikking}
    arrKode      : Array[1..100] of Char;
    arrWaarde    : Array[1..100] of Real;
    dumTyd       : TDateTime;
    dumKode      : Char;
    dumWaarde    : Real;
begin
  //Vertoon die verslag van baba X?
    sNom := Uppercase(edtNommer.Text);
    IF NOT Geldige_Rekening(edtNommer.Text)✓{roep funksie}
     then
      begin
        ShowMessage('Die pasiente nommer is ongeldig.');✓
        Exit; ✓{alternatiewelik met 'n else...begin..end}
      end;
 //Vertoon PERSOONLIKE INFO verslag
    redVerslag.Clear; ✓
    redVerslag.Lines.Add('PASIENT NOMMER: ' + sNom);
    Delete(sPers_Info, 1, pos(',', sPers_Info));
```

✓ pos
✓ copy tot voor komma (p-1)✓
✓ delete

✓ repeat vir al die velde

```
   sV := copy(sPers_Info, 1, pos(',', sPers_Info)-1);
   Delete(sPers_Info, 1, pos(',', sPers_Info));
   sN := copy(sPers_Info, 1, pos(',', sPers_Info)-1);
   Delete(sPers_Info, 1, pos(',', sPers_Info));
   sB := copy(sPers_Info, 1, pos(',', sPers_Info)-1);
   Delete(sPers_Info, 1, pos(',', sPers_Info));
   sG := sPers_Info[1];
   redVerslag.Lines.Add('Pasient: ' + sN + ', ' + sV ); ✓{vertoon info in rich edit}
   redVerslag.Lines.Add('Geboortedatum: ' + sB);

//vertoon verslag vir info
   AssignFile(TF, 'ICU.txt');✓
   Reset(TF);✓
   Tel := 0;✓
   While NOt EOF(TF) DO✓
   begin
     Readln(TF, sLyn); ✓
     IF Pos(sNom, sLyn) > 0✓
      then
       begin
         Inc(Tel, 1);✓
         arrTyd[Tel] := StrToTime✓(copy(slyn, 1, pos(',', sLyn)-1));
         Delete(slyn, 1, pos(',', slyn)); //time
         Delete(sLyn, 1, pos(',', sLyn)); //rek nom
         arrKode[Tel] := slyn[1]; ✓
         Delete(slyn, 1, 2); //kode
         arrWaarde[Tel] := StrToFloat(slyn); ✓//value
       end;
   end;
   CloseFile(TF);✓

   //Sorteer die waardes
   For A := Tel downto 2 do
     For B := 1 to (a-1) do
      IF arrTyd[b] > arrTyd[b+1]
       then
        begin
          dumTyd      := arrTyd[b];
          arrTyd[b]   := arrTyd[b+1];
          arrTyd[b+1] := dumTyd;

          dumKode     := arrKode[b];
          arrKode[b]  := arrKode[b+1];
          arrKode[b+1]:= dumKode;

          dumWaarde      := arrWaarde[b];
          arrWaarde[b]   := arrWaarde[b+1];
          arrWaarde[b+1] := dumWaarde;
        end;
```

Sort = ✓✓✓✓✓
-1 vir elke fout

Sortering

```
   //Display info in report
   //Opskrifte
   redVerslag.Paragraph.TabCount := 3;
   redVerslag.Paragraph.Tab[0]   := 100;
   redVerslag.Paragraph.Tab[1]   := 200;
   redVerslag.Paragraph.Tab[2]   := 300;
   redVerslag.Paragraph.Tab[3]   := 400;
   redVerslag.Lines.Add('');
   redVerslag.SelAttributes.Style := [fsBold];
   redVerslag.Lines.Add('TYD' + #9 + #9 + 'LESINGS');
   redVerslag.SelAttributes.Style := [fsBold];
   redVerslag.Lines.Add('   '+#9+'TEMP' + #9 + 'HART' + #9 + 'SUURSTOF');
   redVerslag.Lines.Add('   '+#9+'°C' + #9 + 'bpm' + #9 + 'ppm');

   For a✓ := 1 to Tel✓ do
   begin
      sLyn := FormatDateTime✓('hh:mm', arrTyd[a]);
      ✓case arrKode[a] of
        'T' : sLyn := sLyn✓ +#9 ✓{kolomme} + FloatToStrF(arrWaarde[a],ffFixed,0,2);✓
        'H' : sLyn := sLyn +#9+#9✓ + FloatToStrF(arrWaarde[a],ffFixed,8,0);
        'O' : sLyn := sLyn +#9+#9+#9✓ + FloatToStrF(arrWaarde[a],ffFixed,8,0);
      end;
      redVerslag.Lines.Add(sLyn); ✓
   end;
```

✓ {*kolomme*}

✓✓ {*opskrifte*}

```
end;

procedure TVRAAG_3.FormCreate(Sender: TObject);
begin
   dlgSaveAs.InitialDir := ExtractFilePath(Application.ExeName);
end;

function TVRAAG_3.Geldige_Rekening(sNomX: String): Boolean;
var
  sNom, sLyn : String;
  TF         : TextFile; ✓
  Gevind     : Boolean;
begin
  //vraag 3.2.
  sNom := Uppercase✓(edtNommer.Text);
  IF (length(sNom) <> 4)✓ AND✓
     NOT((sNom[1] in ['A'..'Z']) AND (sNom[2] in ['A'..'Z'])) ✓✓ AND
     NOT((sNom[3] in ['0'..'9']) AND (sNom[4] in ['0'..'9'])) ✓✓
   then
    begin
      Result := False; ✓
      Exit; ✓ {of met 'n else...begin...end}
    end;
    Gevind := False; ✓
    AssignFile(TF, 'INFO.TXT'); ✓
    Reset(TF); ✓
    While NOT EOF(TF)✓ AND NOT Gevind ✓ DO
     begin
       Readln(TF, sLyn);✓
       sLyn := Uppercase(sLyn);
       IF Copy(sLyn, 1, 4)✓ = sNom✓
        then
         begin
           Gevind := True;✓
           sPers_Info := sLyn; ✓
         end;
     end;
    CloseFile(TF); ✓
    IF NOT Gevind ✓
    then
       Result := False✓
     else
   Result := True; ✓
end;

end.
```

ADDENDUM A

KANDIDAAT: .....

VRAAG	BESKRYWING	MAKS PUNT	LEERDER PUNT	MOD
1	Databasis en Programmering	40		
2	OOP	44		
3	Delphi	36		
	TOTAAL	**120**		

Leerder:
Ek, _____ verklaar hiermee dat die eksamenskrif my eie, oorspronklike werk is.

Leerder: _____ Datum:

Onderwyser: _____ Datum:

Groepmoderator: _____ Datum: _____

ADDENDUM B

VRAAG 1: DELPHI – DATABASIS EN PROGRAMMERING
Nasienrubriek

KANDIDAAT: .....

Vraag	Konsepte	Maks punt	Leerder punt	MOD
1.1.	SELECT Dokter, Tarief (1) FROM Dokters (1) ORDER BY (1) Dokter DESC (1)	4		
1.2.	{SELECT Van, Voornaam, Geb, Gewig FROM Babas} (1) WHERE (YEAR(Geb) (1) = ' + Jaar (1) + ') AND(1) (Gewig < 3.5)(1)	6		
1.3.	SELECT Count(*) (1) AS [Seuns gebore in Jan 2008] (1) FROM Babas WHERE (Geslag = "M") (1) AND (Year(Geb) = 2008) AND (Month(Geb) = 1) (1) {enige korrekte datum manier}	4		
1.4.	UPDATE Babas (1) SET (1) Geb = #2007/06/16# (1) WHERE (Van = "Adam") AND (Voornaam = "Stanley") (1) *Vertoon van rekord (1):* SELECT * FROM Babas WHERE (Van="Adam") AND (Voornaam="Stanley")	5		
1.5.	SELECT Voornaam, Geb, Dokter, (1) Format (1) (DaeInHosp * Tarief (1), "Currency") AS [Koste per kind] (1) FROM Dokters, Babas (1 *beide tabelle*) WHERE (Van = "Green")(1) AND (DokterID = PraktykNom)(1 *linking tabelle*)	7		
1.6.	SELECT TOP 5 Babas.Van, Voornaam, Geb AS [Geboortedatum], Woonbuurt (1) FROM Ouers, Babas WHERE (Ouers.RekNom = Babas.RekNom) (1 *linking tabelle*) AND (Geslag = "F")(1) **AND** (1) ((Woonbuurt="Sinoville")(1) **OR** (1) (Woonbuurt="Rooihuiskraal")) (Die OR moet in hakkies wees.) ORDER BY Geb DESC (1)	7		
1.7.	INSERT INTO Dokters (1) (PraktykNom, Dokter, Tarief) (1 veldname) VALUES ("BEL763", "BELL G", 550.00) (1 waardes) *Vertoon alfabetiese lys* (1): Select * FROM Dokters Order BY Dokter	4		
1.8.	{SELECT Van, Voornaam FROM Babas (1)} WHERE (Voornaam Like "Jo%" (1)) AND (Geslag = "M") (1)	3		
	SINTAKSFOUTE: Penaliseer slegs as fout lei tot 'n logiese fout. (-1 per fout) Maksimum van 2 punte.			
	SUBTOTAAL	**40**		

ADDENDUM C

VRAAG 2: DELPHI – OOP
Nasienrubriek

KANDIDAAT: .....

Vraag	Konsepte	Maks punt	Leerder punt	MOD
2.1.	**CLASS UNIT**			
	Type definisie (1); private velde (1 – string; 1 – real; 1 – Boolean); ***Constructor***: Parameters (2); toeken aan velde (2); IF fisio (1) then (1) else (1); herhaling van IF (1); bepaal balans (2) **Accessors**: GetBalans (2); GetVolleNaam (2)	18		
2.2.	**FORM UNIT**			
	Uses unit (1); Verklaar Array(1) of TPasient(1); teller vir array(1)	4		
2.2.1	**REGISTRASIE**			
	Toets maks 50 pasiënte (2); Toets vir een behandeling (3); Toets naam (2); Inc(aantal) (1); skep array index object (4){N[a] ✓ := TPasient.Create✓(parameters✓✓)}	12		
2.2.2.	**ONTSLAG**			
a.	**NAAMLYS:** Clear richedit (1); Kolomme (1); Opskrifte (1); For-loop tot aantal(1); R.Lines.Add (1); Nr (1); X.GetVolleNaam (1); Format (1 = regs inlyn); twee desimale (1); X.GetBalans (1); Vertoon Aantal op lbl.Caption (1)	11		
b.	**VERWYDER:** VAR Textfile (1); Toets geldige interval (4); Toets numeriese data (4) Exit/Else Begin..End(1) Case ItemIndex (1) **OPSIE WAAKEENHEID:** (1) AssignFile(1); Toets of lêer bestaan (1); Append(1); else (1) Rewrite(1); Boodskap (1); Skryf info na textfile (1); CloseFile(1); **OPSIE ONTSLAAN:** (1) boodskap (1); **VERWYDER UIT SKIK:** X[a].Free (1); for loop index tot aantal (1); skuif elemente op N[a]✓ := N[a+1✓] (2); Dec(Aantal) (1); Boodskap display (1); Click Naamlys-button (1)	29		
2.3.	**VERLAAT OPSIE**			
	Drie global tellers > Inc(oorplaas) (1); Inc(ontslaan) (1); bepaal totaal (2); **BOODSKAP**: Tydformaat (2); oop reël (1); subopskrifte (1); Currency (1); twee desimale (1); vertoon boodskap (1) **VERWYDER ALLE OBJECTS**: For-loop (1); N[a].Free ✓(1) **APPLICATION.TERMINATE** (1)	14		
	SINTAKSFOUTE: Penaliseer slegs as fout lei tot 'n logiese fout. (-1 per fout) tot 'n maksimum van 2 punte.			
	SUBTOTAAL			

ADDENDUM D

VRAAG 3: DELPHI
Nasienrubriek

KANDIDAAT: .....

Vraag	Konsepte	Maks punt	Leerder punt	MOD
3.2.	**FUNCTION GELDIG**:			
	Var Textfile (1); Uppercase(1); toets voorwaardes: lengte van nommer (1), AND (1), karakter 1&2 alfabet (2), karakter 3&4 syfers(2)); Ongeldig result = false (1); Exit(1 of met else); vlag (1); AssignFile(1); Reset(1); While lus met begin...end (2); Readln(1); Copy (eerste 4 letters) (1); IF toets = (1); Vlag true(1); sPers_Info(1); CloseFile(1); Nie gevind (1); Result (2)	24		
3.3.	**VERSLAG**			
3.3.1.	Gebruik funksie en toets (3)	3		
3.3.2.	**VERTOON PERSOONLIKE INFO**: Clear richedit (1); Pos (1); Copy (1); (copy tot voor komma: p-1)(1) ; Delete(1); herhaal vir al die velde (1); Add in richedit (1)	7		
3.3.3.	**VERTOON DATA CHRONOLOGIES**: A) **LEES TEXTFILE INTO ARRAY**: {***11***} Declare array (1) AssignFile (1); Reset (1); Teller init (1); While-loop (1); Readln uit textfile (1); Toets of nommer deel van lyn is (1); Indien nommer in lyn voorkom: Inc teller (1); StrToTime(1); arrKode(1); arrWaarde(1); CloseFile(1); B) **SORTEER ARRAY's**: {***5***} Sorteer al 3x die skikkings (5) C) **VERTOON INFO**: {***12***} Kolomme (1); Opskrifte (2); For-lus (2); FormatDateTime funksie (1); Case arrKode[x] (1); slyn := slyn (1); Aantal #9 {tab's} in lyn bygevoeg (3); array waarde (1); Lines.Add (1)	30		
3.4.	**SAVE AS...**: IF (1); dlg.Execute (1); Kry lêernaam (1); SaveToFile(1)	4		
3.5.	**PRINT**: IF (1);dlg.Execute (1); richedit.Print (2);	4		
	SINTAKSFOUTE: Penaliseer slegs as fout lei tot 'n logiese fout. (-1 per fout) tot 'n maksimum van 2 punte.			
	SUBTOTAAL			

ALTERNATIEWE MEMO: VRAAG 3
Hier is 'n alternatiewe memo vir Vraag 3 wat gebruik maak van classes en objects.
Here is an alternative memorandum for Question 3, making use of classes and objects.

```
unit VR3_CLASS_UNIT_MEMO_U;
//*********************************************
//**  ALTERNATIEWE MEMORANDUM VIR VRAAG 3  **
//**  Class                                **
//*********************************************

interface

USES SysUtils;

TYPE

  TWaarneming = class(TObject)
    private
      fTyd    : TDateTime;
      fKode   : Char;
      fMeting : Real;
    public
      constructor Create;
      procedure SetLesing(Tyd : TDateTime; Kode : Char; Meting : Real);
      function GetTyd : TDateTime;
      function ToString : String;
  end;

   TBaby = class(TObject)
     private
        fNaam     : String;
        fRekNom   : String;
        fGebDat   : String;
        fGender   : String;
        fLesings  : Array[1..100] of TWaarneming;
        fTeller   : Integer;
        procedure SetLesingX(Index : Integer; LX : TWaarneming);
     public
       constructor Create(Line: String);
       procedure SetLesings(sLine : String);
       function GetLesingX(Index : Integer) : TWaarneming;
       function GetNaam : String;
       function GetRekeningNom : String;
       function GetDOB : String;
       function GetTeller : Integer;
       procedure Sorteer;
       procedure StelVry;
   end;


implementation

{ TPasient }

constructor TBaby.Create(Line: String);
var
  a : Integer;
  sX, sY : String;
begin
   fRekNom := Copy(Line, 1, pos(',', Line)-1);
   Delete(Line, 1, pos(',', Line));
   sX := Copy(Line, 1, pos(',', Line)-1);
   Delete(Line, 1, pos(',', Line));
   sY := Copy(Line, 1, pos(',', Line)-1);
   Delete(Line, 1, pos(',', Line));
   fNaam   := sY + ' ' + sX;
   fGebDat := Copy(Line, 1, pos(',', Line)-1);
   Delete(Line, 1, pos(',', Line));
   fGender := Line[1];
   fTeller := 0;
      For a := 1 to 100 do
     fLesings[a] := TWaarneming.Create;
end;
```

```
procedure TBaby.StelVry;
var
   a : integer;
begin
  for a := 1 to 100 do
   self.fLesings[A].Free;
  self.Free;
end;

function TBaby.GetDOB: String;
begin
  result := fGebDat;
end;

function TBaby.GetLesingX(Index: Integer):
TWaarneming;
begin

//*********************************************
procedure TBaby.SetLesings(sLine: String);
var
  st, sk, sl : String;
begin
 // cop/pos
 st := copy(sLine, 1, pos(',', sLine)-1);
 Delete(sline,1, pos(',', sline));
 Delete(sline,1, pos(',', sline)); //acc nom
 sk := copy(sLine, 1, pos(',', sLine)-1);
 Delete(sline,1, pos(',', sline)); //code
 sl := sLine;
 Inc(fTeller, 1);
 fLesings[fTeller].SetLesing(StrToTime(st), sk[1], StrToFloat(sL));
end;

procedure TBaby.SetLesingX(Index: Integer; LX: TWaarneming);
begin
   fLesings[Index] := LX;
end;

procedure TBaby.Sorteer;
var
   a, b  : integer;
   Dummy : TWaarneming;
begin
  For A := Self.fTeller downto 2 do
    for B := 1 to (a-1) do
     IF (Self.GetLesingX(B).GetTyd > Self.GetLesingX(B+1).GetTyd)
      then
       begin
         Dummy := Self.GetLesingX(B);
         Self.SetLesingX(B, Self.GetLesingX(B+1));
         Self.SetLesingX(B+1, Dummy);
       end;
end;

{ TWaarneming }

constructor TWaarneming.Create;
begin
    fTyd    := 0;
    fKode   := #0;
    fMeting := 0.00;
end;

procedure TWaarneming.SetLesing(Tyd: TDateTime; Kode: Char; Meting: Real);
begin
    fTyd    := Tyd;
    fKode   := Kode;
    fMeting := Meting;
end;

function TWaarneming.ToString: String;
var
   sDum : String;
begin
   Result := fLesings[Index];
end;

function TBaby.GetNaam: String;
begin
   Result := fNaam;
end;

function TBaby.GetRekeningNom: String;
begin
  Result := fRekNom;
end;

function TBaby.GetTeller: Integer;
begin
  result := fTeller;
end;
```

```
function TWaarneming.GetTyd: TDateTime;
begin
  Result := fTyd;
end;
```

```
   sDum := FormatDateTime('hh:mm', Self.fTyd);
   Case Self.fKode of
      'T' : sDum := sDum + #9 + Format('%0.2f', [Self.fMeting]);
      'H' : sDum := sDum + #9+#9 + Format('%0.0f', [Self.fMeting]);
      'O' : sDum := sDum + #9+#9+#9 + Format('%0.0f', [Self.fMeting]);
   End;
   Result := sDum;
end;

end.
```

```
unit VR3_CLS_MEMO_2_U;
//*********************************************
//**  ALTERNATIEWE MEMORANDUM VIR VRAAG 3  **
//**   wat gebruik maak van classes & objects! **
//*********************************************
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Buttons, ExtCtrls, ComCtrls,
  VR3_CLASS_UNIT_MEMO_U;
type
  TVRAAG_3 = class(TForm)
  private
    { Private declarations }
    Baby : TBaby;
    sPers_Info                      : String;
    function Geldige_Rekening(sNomX : String) : Boolean;
  public
    { Public declarations }
  end;

var
  VRAAG_3: TVRAAG_3;
implementation
{$R *.dfm}

procedure TVRAAG_3.BitBtn1Click(Sender: TObject);
begin

  Application.Terminate;
end;

procedure TVRAAG_3.bmbPrintClick(Sender: TObject);
begin
  //Maak 'n drukstuk van verslag van baba X?
  IF dlgPrint.Execute
  then redX.Print('Verslag');
end;

procedure TVRAAG_3.bmbSaveAsClick(Sender: TObject);
begin
  //Stoor die verslag van baba X?
 if dlgSaveAs.Execute
   then  redX.Lines.SaveToFile(dlgSaveAs.FileName);
end;

procedure TVRAAG_3.bmbVerslagClick(Sender: TObject);
var
   sLyn, sX : String;
   TF       : TextFile;
   A        : Integer;
begin
  //Vertoon die verslag van baba X?
  sx := uppercase(edtNommer.Text);
  if not Geldige_Rekening(sx)
   then
   begin
     ShowMessage('Rekeningnommer is ONGELDIG.');
     Exit;
   end;
   //valid >> display info
   //STEP 1: PUT ALL INFO INTO THE CLASS ARRAY;
    Baby := TBaby.Create(sPers_Info);
```

```
    AssignFile(TF, 'ICU.txt');
    Reset(TF);
    While NOT EOF(TF) DO
    Begin
      readln(tf, sLyn);
      IF pos(Baby.GetRekeningNom, sLyn) > 0
       then
        begin
          Baby.SetLesings(sLyn);
        end;
    End;
    CloseFile(TF);
   //STEP 2: sort CLASS array
   Baby.Sorteer;
   //Step 3: Display array
   redX.Lines.Clear;
   redX.Lines.Add('ACCOUNT NUMBER: ' + Baby.GetRekeningNom);
   redX.Lines.Add('BABY          : ' + Baby.GetNaam);
   redX.Lines.Add('DATE OF BIRTH : ' + Baby.GetDOB);
   redX.Lines.Add(' ');
   redX.Paragraph.TabCount := 3;
   redX.Paragraph.Tab[0] := 100;
   redX.Paragraph.Tab[1] := 200;
   redX.Paragraph.Tab[2] := 300;
   redX.Lines.Add('TIME' + #9 + #9 + 'MEASUREMENTS');
   redX.Lines.Add(#9 + 'TEMP' + #9 + 'HEART' + #9 + 'OXYGEN');
   redX.Lines.Add(#9 + '°C'   + #9 + 'bpm'   + #9 + 'ppm');
   for A := 1 to Baby.GetTeller do
    redx.Lines.Add(Baby.GetLesingX(A).ToString);
end;

procedure TVRAAG_3.FormClose(Sender: TObject; var Action: TCloseAction);
begin
   Baby.StelVry;
end;

procedure TVRAAG_3.FormCreate(Sender: TObject);
begin
   dlgSaveAs.InitialDir := ExtractFilePath(Application.ExeName);
end;
```

```
function TVRAAG_3.Geldige_Rekening(sNomX:
String): Boolean;
var
  sNom, sLyn : String;
  TF         : TextFile;
  Gevind     : Boolean;
begin
  //vraag 3.2.
  sNom := Uppercase(edtNommer.Text);
  IF (length(sNom) <> 4) AND
     NOT((sNom[1] in ['A'..'Z']) AND
(sNom[2] in ['A'..'Z'])) AND
     NOT((sNom[3] in ['0'..'9']) AND
(sNom[4] in ['0'..'9']))
   then
    begin
      Result := False;
      Exit;
    end;
    Gevind := False;
    AssignFile(TF, 'INFO.TXT');
end.
```

```
    Reset(TF);
    While NOT EOF(TF) AND NOT Gevind DO
     begin
       Readln(TF, sLyn);
       sLyn := Uppercase(sLyn);
       IF Copy(sLyn, 1, 4) = sNom
        then
         begin
           Gevind := True;
           sPers_Info := sLyn;
         end;
     end;
    CloseFile(TF);
    IF NOT Gevind
    then
     begin
       Result := False;
       Exit;
     end;
   Result := True;
end;
```