



Province of the  
**EASTERN CAPE**  
EDUCATION

**NATIONAL  
SENIOR CERTIFICATE**

**GRADE/GRAAD 12**

**SEPTEMBER 2013**

**INFORMATION TECHNOLOGY P1 /  
INLIGTINGSTEGNOLOGIE V1  
MEMORANDUM**

**MARKS/PUNTE: 120**

---

This memorandum consists of 12 pages./  
Hierdie memorandum bestaan uit 12 bladsye.

---

**QUESTION/VRAAG 1: DATABASE AND DELPHI****(33 marks/punte)**

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  qryschool.Active := false;
  qryschool.SQL.text := 'select *✓ from childrenTb✓ order by name✓';
  qryschool.Active := true ;
  SetGridColumnWidths(DBGrid1);
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  qryschool.Active := false;
  qryschool.SQL.text := 'select name✓ from childrenTb✓ where (allergies = true) ✓ and (ExtraActivities like "%Cooking is Fun%")✓';
  qryschool.Active := true;
  SetGridColumnWidths(DBGrid1);
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  qryschool.Active := false;
  qryschool.SQL.text := 'select name✓, (2013 - year(dateOfBirth) ✓) as [Age] ✓ from childrenTb✓';
  qryschool.Active := true;
  SetGridColumnWidths(DBGrid1);
end;
```

```
procedure TForm1.Button4Click(Sender: TObject);
begin
  qryschool.Active := false;
  qryschool.SQL.text := 'select name,dad_name,mom_name✓ from childrenTb✓,parentsTb✓ where childrenTb.ChildID = parentsTb.childID ✓and AvailableToHelp = true✓';
  qryschool.Active := true;
  SetGridColumnWidths(DBGrid1);
end;
```

```
procedure TForm1.Button5Click(Sender: TObject);
begin
  qryschool.Active := false;
  qryschool.SQL.text := 'select count(*)✓ as [Boys doing Ball Skills] ✓ from childrenTb✓ where ExtraActivities like "%Ball Skills%"✓';
  qryschool.Active := true;
  SetGridColumnWidths(DBGrid1);
end;
```

```
procedure TForm1.Button6Click(Sender: TObject);
begin
  qryschool.Active := false;
  qryschool.SQL.text := 'insert into✓ childrenTb✓ values (23,"Rebekka","White",2009/04/02,"F",No,"Cooking is Fun")✓';
  qryschool.ExecSQL;
  qryschool.SQL.Text := 'select * from childrenTb✓';
  qryschool.Active := true;
```

```
SetGridColumnWidths(DBGrid1);  
end;
```

```
procedure TForm1.Button7Click(Sender: TObject);  
var  
sactivity : string;  
begin  
sactivity := inputbox('Enter activity', "", ""); ✓  
qryschool.Active := false;  
qryschool.SQL.text := 'select name, surname✓ from childrenTb✓ where ExtraActivities like  
"%'+sactivity+'%"✓';  
qryschool.Active := true;  
SetGridColumnWidths(DBGrid1);  
end;
```

```
procedure TForm1.Button8Click(Sender: TObject);  
begin  
qryschool.Active := false;  
qryschool.SQL.text := 'select name, surname, dateofbirth✓ from childrenTb✓ where  
(MONTH(Dateofbirth)=10) ✓ or (MONTH(Dateofbirth)=11) ✓ or (MONTH(Dateofbirth)=12) ✓';  
qryschool.Active := true; *where (MONTH(DateOfBirth) in (10,11,12))  
SetGridColumnWidths(DBGrid1); can also be used  
end;
```

**QUESTION/VRAAG 2****(50 marks/punte)**

|     |       |                                                               |   |
|-----|-------|---------------------------------------------------------------|---|
| 2.1 | 2.1.1 | Define a class TSchool                                        | 1 |
|     |       | Private                                                       | 1 |
|     |       | Declaring fname, fdob, fdeposit, fage                         | 1 |
|     | 2.1.2 | Constructor heading                                           | 1 |
|     |       | Assigning and initialising fields                             | 1 |
|     | 2.1.3 | Procedure CalcAge heading                                     | 1 |
|     |       | Calculate fage                                                | 1 |
|     |       | SysUtils                                                      | 1 |
|     | 2.1.4 | Function getreason : string heading                           | 1 |
|     |       | Nested if statements                                          | 6 |
|     | 2.1.5 | Function accepted : Boolean heading                           | 1 |
|     |       | If deposit paid and age is correct then accept should be true | 3 |
|     | 2.1.6 | GetName heading                                               | 1 |
|     |       | fname assigned to result/function                             | 1 |
|     | 2.1.7 | GetDOB heading                                                | 1 |
|     |       | fdob assigned to result/function                              | 1 |
|     |       | /23/                                                          |   |
| 2.2 | 2.2.1 | Declare arrlearners, icount (global)                          | 2 |
|     |       | Initialise counter                                            | 1 |
|     |       | Check if file exists, ShowMessage                             | 2 |
|     |       | Assignfile and Reset                                          | 2 |
|     |       | While not eof() do                                            | 1 |
|     |       | Readln                                                        | 1 |
|     |       | Increase counter                                              | 1 |
|     |       | Get name                                                      | 1 |
|     |       | Get dob                                                       | 1 |
|     |       | Get deposit                                                   | 1 |
|     |       | Assign to arrlearner                                          | 1 |
|     | 2.2.2 | Loop                                                          | 1 |
|     |       | Call getname → display                                        | 1 |
|     | 2.2.3 | Loop                                                          | 1 |
|     |       | Call calcage                                                  | 1 |
|     |       | If accepted = true then increase counter and display the name | 3 |
|     |       | Display how many learners have been accepted                  | 1 |
|     | 2.2.4 | Loop                                                          | 1 |
|     |       | Call calcage                                                  | 1 |
|     |       | If accepted = false then display the name and reason          | 3 |
|     |       | /27/                                                          |   |

**POSSIBLE SOLUTION**

```
unit Unit2;
```

```
interface
```

```
uses
```

```
  sysutils;✓
```

```
type
```

```
  TSchool = class✓
```

```
  private✓
```

```
    fname : string;
```

```
    fdob : string;
```

```
    fdeposit : boolean;
```

```
    faccept : boolean;
```

```
    fage : integer;
```

```
  public
```

```
    constructor create(sname, sdob : string; bdeposit : boolean);
```

```
    procedure calcage;
```

```
    function accepted : boolean;
```

```
    function getname : string;
```

```
    function getdob : string;
```

```
    function getreason : string;
```

```
  end;
```

```
implementation
```

```
constructor TSchool.create(sname, sdob : string; bdeposit : boolean); ✓
```

```
begin
```

```
  fname := sname;
```

```
  fdob := sdob;
```

```
  fdeposit := bdeposit;
```

```
  faccept := false;
```

```
end;
```

```
procedure TSchool.calcage; ✓
```

```
begin
```

```
  fage := 2013 - strtoint(copy(fdob,7,4)); ✓
```

```
end;
```

```
function TSchool.getreason : string; ✓
```

```
begin
```

```
  if (fage < 4) or (fage > 6) then✓
```

```
    getreason := 'Incorrect Age'✓
```

```
  else
```

```
    if (fdeposit = false) then✓
```

```
      getreason := 'Deposit not paid'✓
```

```
    else
```

```
      if ((fage < 4) or (fage > 6)) and (fdeposit = false) then✓
```

```
        getreason := 'Deposit not paid and incorrect age'; ✓
```

```
  end;
```

```
function TSchool.accepted : boolean; ✓
```

```
begin
```

```
  if (fdeposit = true) ✓ and ((fage >= 4) and (fage <= 6)) ✓ then
```

```
    result := true; ✓
```

```
end;
```

```
function TSchool.getdob : string; ✓
```

```
begin
  result := fdob; ✓
end;
```

```
function Tschool.getname : string; ✓
begin
  result := fname; ✓
end;
end.
```

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Menus, StdCtrls, ComCtrls, unit2;

type

```
TForm1 = class(TForm)
  MainMenu1: TMainMenu;
  RichEdit1: TRichEdit;
  Displayallcandidates1: TMenuItem;
  Close1: TMenuItem;
  AcceptedCandidates1: TMenuItem;
  ReasonsforNonAcceptance1: TMenuItem;
  DisplayallApplications1: TMenuItem;
  Exit1: TMenuItem;
  procedure DisplayallApplications1Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure Close1Click(Sender: TObject);
  procedure AcceptedCandidates1Click(Sender: TObject);
  procedure ReasonsforNonAcceptance1Click(Sender: TObject);
  procedure Exit1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

```
Form1: TForm1;
icount : integer; ✓
arrschools : array[1..50] of Tschool; ✓
```

implementation

{ \$R \*.dfm }

```
procedure TForm1.DisplayallApplications1Click(Sender: TObject);
```

var

```
k:integer;
```

begin

```
for k := 1 to icount do ✓
```

```
begin
```

```
richedit1.lines.add(arrschools[k].getname); ✓
```

```
end;
```

end;

```
procedure TForm1.FormCreate(Sender: TObject);
```

var

```
myfile : textfile;
```

```
soneline,sname,sdob : string;
```

```
k,ipos: integer;
```

```
bdeposit : boolean;
```

begin

```
icount := 0; ✓
```

```
if fileexists('Applications.txt') <> true then ✓
```

```
begin
```

```
ShowMessage('File does not exist'); ✓
```

```

    Exit;
end;
Assignfile(myfile,'Applications.txt'); ✓
Reset(myfile); ✓
while not eof(myfile) do ✓
begin
    readln(myfile,soneline); ✓
    ipos := pos('#',soneline);
    sname := copy(soneline,1, ipos-1); ✓
    delete(soneline,1,ipos);
    ipos := pos('#',soneline);
    sdob := copy(soneline,1, ipos-1); ✓
    delete(soneline,1,ipos);
    bdeposit := strtobool(soneline); ✓
    inc(icount); ✓
    arrschools[icount]:=Tschool.create(sname, sdob, bdeposit); ✓
end;
closefile(myfile);
end;

procedure TForm1.AcceptedCandidates1Click(Sender: TObject);
var
    k, itotal : integer;
begin
    richedit1.clear;
    for k := 1 to icount do ✓
    begin
        arrschools[k].calcage; ✓
        if arrschools[k].accepted = true then ✓
        begin
            itotal := itotal + 1; ✓
            richedit1.Lines.add(arrschools[k].getname); ✓
        end;
    end;
    richedit1.lines.add(inttostr(itotal)+' children accepted'); ✓
end;

procedure TForm1.ReasonsforNonAcceptance1Click(Sender: TObject);
var
    k : integer;
begin
    richedit1.Clear;
    for k := 1 to icount do ✓
    begin
        arrschools[k].calcage; ✓
        if arrschools[k].accepted = false then ✓
        richedit1.Lines.add(arrschools[k].getname ✓ + #9 + arrschools[k].getreason) ✓
    end;
end;

procedure TForm1.Exit1Click(Sender: TObject);
begin
    Exit;
end;

end.

```



**QUESTION/VRAAG 3****(37 marks/punte)**

|     |                                                                                       |   |
|-----|---------------------------------------------------------------------------------------|---|
| 3.1 | Initialise row counter                                                                | 1 |
|     | Check if file exist and error message                                                 | 1 |
|     | Assignfile and reset                                                                  | 1 |
|     | Readln                                                                                | 1 |
|     | Initialise column counter                                                             | 1 |
|     | Increase row counter                                                                  | 1 |
|     | Loop                                                                                  | 1 |
|     | Assign ticket numbers to array                                                        | 2 |
| 3.2 | For row loop                                                                          | 1 |
|     | For col loop                                                                          | 1 |
|     | Display array in stringgrid                                                           | 1 |
| 3.3 | Initialise counter                                                                    | 1 |
|     | For loops                                                                             | 1 |
|     | If statement to see if a ticket has been sold                                         | 1 |
|     | Increase counter if true                                                              | 1 |
|     | Display how many tickets have been sold                                               | 1 |
| 3.4 | Random(230) + 1                                                                       | 1 |
|     | For loops                                                                             | 1 |
|     | If ticket number = random number, then display in listbox                             | 2 |
|     | Increase counter                                                                      | 1 |
|     | Repeat until counter = 10                                                             | 1 |
| 3.5 | Initialise variables                                                                  | 2 |
|     | For loops                                                                             | 1 |
|     | If a ticket was sold increase counter                                                 | 2 |
|     | If it is more than previous values, then assign highest to max and remember which row | 3 |
|     | Display the child's name who sold the most tickets                                    | 1 |
| 3.6 | Calculate amount                                                                      | 1 |
|     | Calculate amount to be donated                                                        | 1 |
|     | Display amount and donated amount                                                     | 2 |
|     | Rounded to two decimal places                                                         | 1 |

**POSSIBLE SOLUTION**

unit Question3\_u;

interface

uses

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

type

TForm1 = class(TForm)

...

end;

var

Form1: TForm1;

ar2tickets : array[1..23,1..10] of integer;

iticketssold : integer;

const

arrnames : array[1..23] of string = ('Jack','Sarah','Ben','Lily','Jocelyn','Adrian',  
                                          'Anathi','Anita','Ashwin','Peter','Chloe','Jeff','Nathi',  
                                          'Wandile','Kerry','Sethu','Caryn','Nicola','Phila',  
                                          'Lindelwa','Kevin','Alex','Rebekka');

implementation

{ \$R \*.dfm }

procedure TForm1.FormCreate(Sender: TObject);

var

irow : integer;

begin

for irow := 1 to 23 do

    stringgrid1.cells[0,irow-1] := arrnames[irow];

end;

procedure TForm1.Button1Click(Sender: TObject);

var

irow,icol,ipos : integer;

myfile : textfile;

soneline : string;

begin

irow := 0; ✓

if fileexists('Tickets.txt') <> true then

begin

    ShowMessage('File does not exist'); ✓

    Exit;

end;

Assignfile(myfile, 'Tickets.txt'); ✓

Reset(myfile);

while not eof(myfile) do

begin

```
readln(myfile, soneline); ✓  
icol := 0; ✓  
inc(irow); ✓  
ipos := pos(',',soneline);  
while ipos <> 0 do ✓  
  begin  
    inc(icol);  
    ar2tickets[irow,icol] := strtoint(copy(soneline,1,ipos-1)); ✓✓  
    delete(soneline,1,ipos);  
    ipos := pos(',',soneline);  
  end;  
end;  
closefile(myfile);
```

```
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);  
var  
  irow, icol : integer;  
begin  
  for irow := 1 to 23 do ✓  
    for icol := 1 to 10 do ✓  
      stringgrid1.cells[icol,irow-1] := inttostr(ar2tickets[irow,icol]); ✓  
    end;  
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);  
var  
  irow, icol : integer;  
begin  
  iticketssold := 0; ✓  
  for irow := 1 to 23 do ✓  
    for icol := 1 to 10 do  
      if ar2tickets[irow,icol] > 0 then ✓  
        inc(iticketssold); ✓  
      ShowMessage(inttostr(iticketssold) + ' tickets sold'); ✓  
    end;  
end;
```

```
procedure TForm1.Button4Click(Sender: TObject);  
var  
  iran, irow, icol, idraws : integer;  
begin  
  randomize;  
  repeat  
    iran := random(230) + 1; ✓  
    for irow := 1 to 23 do ✓  
      begin  
        for icol := 1 to 10 do  
          begin  
            if ar2tickets[irow,icol] = iran then ✓  
              begin  
                listbox1.Items.add(inttostr(iran)) ✓;  
                inc(idraws); ✓  
              end  
            end  
          end;  
        end;  
      end;  
    end;  
  end;
```

```
    end;
    until idraws = 10; ✓
end;

procedure TForm1.Button5Click(Sender: TObject);
var
    imax, irow,icol,ihighest,ichild : integer;

begin
    imax := 0; ✓
    for irow := 1 to 23 do ✓
        begin
            ihighest := 0; ✓
            for icol := 1 to 10 do
                begin
                    if ar2tickets[irow,icol] <> 0 then ✓
                        inc(ihighest); ✓
                    end;
                end;
            if ihighest > imax then ✓
                begin
                    imax := ihighest; ✓
                    ichild := irow; ✓
                end;
            end;
        end;
        ShowMessage(stringgrid1.cells[0,ichild-1] + ' sold the most tickets'); ✓
    end;

procedure TForm1.Button6Click(Sender: TObject);
var
    ramount, rdonate : real;

begin
    ramount := iticketssold * 25; ✓
    rdonate := 40/100*ramount; ✓
    ShowMessage('Total Amount Raised '+floattostrf(ramount, fffixed,8,2)); ✓
    ShowMessage('Total Amount Donated '+floattostrf(rdonate, fffixed,8,2)) ✓✓
end;

end.
```