Province of the
EASTERN CAPE
EDUCATION

NATIONAL SENIOR CERTIFICATE

GRADE 12

SEPTEMBER 2016

INFORMATION TECHNOLOGY P1
INLIGTINGSTEGNOLOGIE V1
MEMORANDUM

MARKS/*PUNTE*: 150

This memorandum consists of 19 pages/
Hierdie memorandum bestaan uit 19 bladsye.

SECTION/*AFDELING* A

QUESTION 1: MARKING GRID – GENERAL PROGRAMMING SKILLS

QUESTION *VRAAG*	DESCRIPTION *BESKRYWING*	MAX MARKS	LEARNER'S MARKS
1.1	**Button – Question 1.1** Get learner's name and surname from text box ✓ Extract initial ✓ Extract surname ✓ Get grade ✓ Get subjectFrom and subjectTo ✓ If subjectFrom = subjectTo ✓ Display 'Incorrect Information' message ✓ Else ✓ If nothing was selected ✓ Display 'learners initial and surname, grade and No subject change' message ✓ Else ✓ Display initial, surname, grade and subjects ✓ Use a showmessage to display message that documents are required ✓	13	
1.2	**Button – Question 1.2** Get date of birth (YYYY/MM/DD) ✓ Get ID ✓ If year, ✓ month ✓ and day ✓ is the same Display Correct message ✓ Else ✓ Display Incorrect message ✓	8	
1.3	**Button – Question 1.3** Get name (either from text area or class scope) ✓ Get length of name ✓ Initialise counter ✓ Conditional Loop ✓ Increment counter ✓ Check if vowel ✓ or space ✓ and create new variable ✓ Generate random number between 100 and 300 (inclusive) ✓ Create username: name ✓ +%+number+# ✓	11	

QUESTION *VRAAG*	DESCRIPTION *BESKRYWING*	MAX MARKS	LEARNER'S MARKS
1.4	**Button – Question 1.4** Initialise counters ✓ Get marks from spinedits ✓ Loop through all marks ✓ Check which subjects are above 40 ✓ Increase counter ✓ Check which subjects are above 30 ✓ Increase counter ✓ Check which subjects are below 30 ✓ Increase counter ✓ If (HomeLanguage is below 40) ✓ or (3 subjects below 40) ✓ or (more than 1 below 30) ✓ Result is Fail ✓ Else ✓ Result is Pass ✓	15	
	TOTAL SECTION/*TOTAAL AFDELING* A:	**47**	

SECTION/*AFDELING* B

QUESTION 2: MARKING GRID – OBJECT-ORIENTATED PROGRAMMING

QUESTION *VRAAG*		DESCRIPTION *BESKRYWING*	MAX MARKS	LEARNER'S MARKS
2.1	2.1.1	**Type declaration** Sysutils ✓ Array declaration – (max 10, integer) ✓ correct ✓ Public Methods constructor create(sname: string; isubject:integer); ✓ procedure APS(arrmarks: Tarrmarks); ✓ function bachpass(arrmarks : Tarrmarks):boolean; ✓ function tostring: string; ✓	7	
	2.1.2	**Constructor** Assign fname, ✓ fsubject (integer)✓ Initialise fapspoints to 0 ✓	3	
	2.1.3	**APS** Loop till fsubjects ✓ Get the levels for different percentages ✓ using the array ✓ and add to fapspoints ✓	4	
	2.1.4	**Bachpass** Initialise counter ✓ Loop till fsubjects ✓ If the mark is greater or equal to 50 ✓ Increase counter ✓ If counter is greater or equal to 4 ✓ Result is true ✓ Else ✓ Result is false ✓	8	
	2.1.5	**Compilestring** Labels ✓ Information on separate lines ✓ Variables fname and fapspoints ✓	3	

QUESTION *VRAAG*	DESCRIPTION *BESKRYWING*	MAX MARKS	LEARNER'S MARKS
2.2	Get name ✓ Get number of subjects ✓ Check if number of subjects is 7 or more ✓ If not ✓ ask user to enter ✓ until it is 7 or more ✓ Create object ✓ with correct parameters ✓ For Loop ✓ till fsubjects ✓ Get marks ✓ and assign to array ✓ If Learner has bachelor pass ✓ Calculate APS points ✓ Display tostring method ✓ Else ✓ Display message that Bachelor Pass requirements have not been met ✓	17	
	TOTAL SECTION/*TOTAAL AFDELING* B:	**42**	

SECTION C

QUESTION 3: MARKING GRID – PROBLEM-SOLVING PROGRAMMING

QUESTION *VRAAG*	DESCRIPTION *BESKRYWING*	MAX MARKS	LEARNER'S MARKS
Program techniques	**Dynamic Component** ✓✓ **Modular Design** • Define and create at least one method/function/procedure and use it correctly ✓ **Programming Techniques** • Descriptive variable names ✓ • Indentation ✓	5	
Get Information, Sort and Display	**Get information from Text File – examnumbers.txt** Test if file exists ✓ Assignfile ✓ Reset ✓ Initialise counter ✓ Loop through file ✓ Read each line from text file ✓ Increment counter ✓ Assign info into array ✓ Close the file ✓ If the file does not exist display a suitable message ✓ **Sort the array** Outer loop ✓ Inner loop ✓ Compare the two index position of array ✓ Swop if needed ✓✓✓ **Display** Loop ✓ Display the contents of array in richedit ✓	18	

Search	Initialise variables ✓ Get input to search ✓ Conditional Loop with correct conditions ✓ bfound:= false; If input is found in array ✓ Copy the school name ✓ Increase index counter ✓ If input was found Display the school name ✓ Loop ✓ Increase index counter ✓ Copy the centre ✓ If the centre is the same ✓ Increase counter ✓ Display the exam number ✓ Display the total learners from that exam centre ✓ If not found, display a message ✓	15	
Add	Get centre and exam number to be added ✓ Increase the counter ✓ Add the new exam number to array arrexamnumbers ✓ Append to textfile ✓ Write to file ✓ Close the file ✓ Display a message that exam number has been added to the file ✓ Sort the array ✓ Clear the richedit ✓ Display the updated array ✓	10	
Delete	Rewrite the 'examnumber.txt' file ✓ Get the exam number to be deleted ✓ Loop ✓ to find the index of exam number in array ✓ Loop from index to be deleted to counter – 1 ✓ Shift contents of array one to left ✓ Decrease counter ✓ Display a message that learner was deleted ✓ Call the Sort method ✓ **Display and write to file** Loop ✓ Write to file ✓ Display contents in richedit ✓ Closefile(myfile); ✓	13	
	TOTAL SECTION/*TOTAAL AFDELING* C:	**61**	
	GRAND TOTAL/*GROOTTOTAAL*:	**150**	

NAME OF LEARNER: _____ GRADE 12: _____

SUMMARY OF LEARNER'S MARKS:

	SECTION A	SECTION B	SECTION C	
	QUESTION 1	QUESTION 2	QUESTION 3	GRAND TOTAL
MAX MARKS	47	42	61	150
LEARNER'S MARKS				

QUESTION 1 – SAMPLE SOLUTION

```
unit Question1_u;

interface

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

type
  TForm1 = class(TForm)
    Panel1: TPanel;
    Edit1: TEdit;
    RadioGroup1: TRadioGroup;
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    RichEdit1: TRichEdit;
    Button1: TButton;
    Label1: TLabel;
    Panel2: TPanel;
    Panel3: TPanel;
    Edit2: TEdit;
    Edit3: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    Button2: TButton;
    Button3: TButton;
    Panel4: TPanel;
    Button4: TButton;
    ComboBox3: TComboBox;
    ComboBox4: TComboBox;
    ComboBox5: TComboBox;
    ComboBox6: TComboBox;
    Edit4: TEdit;
    Edit5: TEdit;
    Edit6: TEdit;
    Edit7: TEdit;
    SpinEdit1: TSpinEdit;
    SpinEdit2: TSpinEdit;
    SpinEdit4: TSpinEdit;
    SpinEdit5: TSpinEdit;
    SpinEdit6: TSpinEdit;
    SpinEdit7: TSpinEdit;
    Panel5: TPanel;
    SpinEdit3: TSpinEdit;
    Panel6: TPanel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
```

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

var
  Form1: TForm1;
  ichange : integer;
  slearner : string;
implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
var
 sinitial, ssurname, ssubjectto,   ssubjectfrom, sgrade : string;
 ipos : integer;
begin
  slearner := edit1.text;
  sinitial := slearner[1];
  ipos := pos(' ',slearner);
  ssurname := copy(slearner,ipos + 1,length(slearner)-ipos);
  sgrade := radiogroup1.items[radiogroup1.itemindex];
  ssubjectfrom := combobox1.text;
  ssubjectto := combobox2.text;
  if ssubjectfrom = ssubjectto then
    richedit1.Lines.add('Incorrect information')
  else
  if (combobox1.ItemIndex = -1) and (combobox2.ItemIndex = -1) then
    richedit1.lines.add('Learner: '+sinitial + ' '+ssurname+#13+'Grade: '+sgrade+#13+'NO
SUBJECT CHANGE')
  else
    begin
      richedit1.lines.add('Learner: '+sinitial + ' '+ssurname+#13+'Grade:
'+sgrade+#13+'Subject Change from '+ ssubjectfrom+ ' to '+ ssubjectto);
      Showmessage('Please submit all the required documents');
      end;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
  sdob, sid : string;
begin
  sdob := edit2.text;
  sid := edit3.text;
  if (copy(sdob,3,2) = copy(sid,1,2)) and (copy(sdob,6,2) = copy(sid,3,2)) and
(copy(sdob,9,2) = copy(sid,5,2)) then
    Showmessage('Correct')
  else
    Showmessage('Do not match');
end;
```



```
procedure TForm1.Button4Click(Sender: TObject);
var
  arrsubjects : array[1..7] of integer;
  iabove40, iabove30, ibelow30, k : integer;
begin
  iabove40 := 0;
  ibelow30 := 0;
  iabove30 := 0;
  arrsubjects[1] := spinedit1.value;
  arrsubjects[2] := spinedit2.value;
  arrsubjects[3] := spinedit3.value;
  arrsubjects[4] := spinedit4.value;
  arrsubjects[5] := spinedit5.value;
  arrsubjects[6] := spinedit6.value;
  arrsubjects[7] := spinedit7.value;

  if arrsubjects[1] < 40 then
    panel5.Caption := 'FAIL';

  for k := 1 to 7 do
   begin
    if arrsubjects[k] >= 40 then
      begin
        inc(iabove40);
      end
    else
    if (arrsubjects[k] >= 30) and (arrsubjects[k] < 40) then
      begin
        inc(iabove30);
      end
    else
      if (arrsubjects[k] < 30) then
        inc(ibelow30);
   end;

  if (arrsubjects[1] < 40) or (iabove40 < 3) or (ibelow30 >=2 )  then
    panel5.Caption := 'FAIL'
  else
    panel5.caption := 'PASS';
end;

procedure TForm1.Button3Click(Sender: TObject);
var
  ilen, k, iran : integer;
  susername : string;
begin
  randomize;
  ilen := length(slearner);
  ShowMessage(inttostr(ilen));
  susername := '';
  k := 0;
  while (k < ilen) do
```

```
    begin
      inc(k);
      if not (upcase(slearner[k]) in ['A','E','I','O','U',' ']) then
        susername := susername + slearner[k];
    end;
  showmessage(susername);
  iran := random(301)+100;
  panel6.Caption := susername + '%' + inttostr(iran) + '#';
end;

end.
```

QUESTION 2 – SAMPLE SOLUTION

```
unit clsLearners;

interface
uses sysutils;
type

TArrmarks = array[1..7] of integer;

TLearner = class
  private
    fname : string;
    fapspoints : integer;
    fbachpass : boolean;
    fsubject : integer;

  public
    constructor create(sname: string; isubject:integer);
    procedure APS(arrmarks : Tarrmarks);
    function bachpass(arrmarks : Tarrmarks):boolean;
    function tostring : string;
end;

implementation

{ TLearner }

procedure TLearner.APS(arrmarks: Tarrmarks);
var
  k, ilevel : integer;
begin
 for k := 1 to 7 do
 begin
   case arrmarks[k] of
     80..100 : ilevel := 7;
     70..79 : ilevel := 6;
     60..69 : ilevel := 5;
     50..59 : ilevel := 4;
     40..49 : ilevel := 3;
     30..39 : ilevel := 2;
     0..29 : ilevel := 1;
   end;
   fapspoints := fapspoints + ilevel;
  end;
end;

function TLearner.bachpass(arrmarks : Tarrmarks): boolean;
var
  ibach, k : integer;
begin
  ibach := 0;
```

```
  for k := 1 to fsubject do
    begin
      if arrmarks[k] >=50 then
        inc(ibach);
    end;
  if ibach >= 4 then
    result := true
  else
    result := false;
end;

constructor TLearner.create(sname: string; isubject:integer);
begin
  fname := sname;
  fsubject := isubject;
  fapspoints := 0;
end;

function TLearner.tostring: string;

begin
   result := 'Learner Name: '+ fname + #13+ 'Bachelor Pass: Yes'+#13+'APS points:
'+inttostr(fapspoints);
 end;

end.

unit Question2_u;

interface

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

type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Button1: TButton;
    Label1: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  Learner : TLearner;
  arrmarks : Tarrmarks;
implementation
```



```
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
var
  sname : string;
  k, isubject : integer;
begin
  sname := edit1.text;

  isubject := Strtoint(inputbox('How many subjects are you registered for?','',''));
  if isubject < 7 then
  begin
  repeat
  Showmessage('Minimum of 7 subjects are required');
  isubject := Strtoint(inputbox('How many subjects are you registered for?','',''));
  until (isubject >=7);
  end;
  learner := TLearner.create(sname, isubject);
  for k := 1 to isubject do
    begin
    arrmarks[k] := strtoint(inputbox('Enter Marks','Subject '+inttostr(k),''));
    end;

  if learner.bachpass(arrmarks) = true then
   begin
    learner.aps(arrmarks);
    Showmessage(learner.tostring);
   end
  else
   Showmessage('You did not meet the Bachelor Pass requirements');
end;

end.
```

QUESTION 3 – SAMPLE SOLUTION

```
unit Question3_u;

interface

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

type
  TForm1 = class(TForm)
    Button1: TButton;
    RichEdit1: TRichEdit;
    Button2: TButton;
    Button3: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Button4: TButton;
    Label1: TLabel;
    Label2: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
  private
    { Private declarations }
  public
    procedure Sort;

  end;

var
  Form1: TForm1;
  icount : integer;
  arrexamnumbers : array[1..250] of string;
  arrcentres : array[1..2] of string = ('4181078,Stirling High School', '4181009,Clarendon
High School for Girls');
implementation

{$R *.dfm}

procedure TForm1.FormCreate(Sender: TObject);
var
  myfile : textfile;
  soneline : string;
begin
  if fileexists('examnumbers.txt') = true then
  begin
    Assignfile(myfile, 'examnumbers.txt');
    Reset(myfile);
```

```
    icount := 0;
    while not eof(myfile) do
    begin
      readln(myfile,soneline);
      inc(icount);
      arrexamnumbers[icount] := soneline;
    end;
    Sort;
    Closefile(myfile);
  end
  else
    Showmessage('File does not exist');
end;

procedure TForm1.Sort;
var
  k, l : integer;
  stemp : string;
begin
  for k := 1 to icount-1 do
    for l := k + 1 to icount do
      begin
        if arrexamnumbers[k] > arrexamnumbers[l] then
          begin
            stemp := arrexamnumbers[k];
            arrexamnumbers[k] := arrexamnumbers[l];
            arrexamnumbers[l] := stemp;
          end;
      end;
end;

procedure TForm1.Button1Click(Sender: TObject);
var
  k : integer;
begin
  for k := 1 to icount do
    richedit1.lines.add(arrexamnumbers[k]);
end;

procedure TForm1.Button2Click(Sender: TObject);
var
  k,j, ipos, ilen, inum : integer;
  ssearch,sschoolname,scentre : string;
  bfound : boolean;
begin
  richedit1.Lines.clear;
  k := 0;
  j := 1;
  inum := 0;
  ssearch := inputbox('','','');
  bfound := false;
  while (j <= 2) and (bfound = false) do
```

```
   begin
    if  copy(arrcentres[j],4,4) = copy(ssearch,4,4) then
     begin
      bfound := true;
      ipos := pos(',',arrcentres[j]);
      ilen := length(arrcentres[j]);
      sschoolname := copy(arrcentres[j],ipos+1,ilen-ipos);
     end;
     inc(j);
   end;

    if bfound = true then
    begin
        richedit1.lines.add(sschoolname + #13);
        while (k <= icount) do
        begin
          inc(k);
          scentre := copy(arrexamnumbers[k],6,4);
          if scentre = copy(ssearch,4,4) then
          begin
            inc(inum);
            richedit1.lines.add(arrexamnumbers[k]);
          end;
        end;
        richedit1.lines.add('Total Learners: ' + inttostr(inum));
    end
   else
    richedit1.lines.add('Centre not found');
end;

procedure TForm1.Button3Click(Sender: TObject);
var
  myfile : textfile;
  scentre, sexamnumber : string;
  k : integer;
begin
  Assignfile(myfile, 'examnumbers.txt');
  Append(myfile);
  scentre := edit1.text;
  sexamnumber := edit2.text;
  inc(icount);
  arrexamnumbers[icount] := sexamnumber;
  writeln(myfile, sexamnumber);
  Closefile(myfile);
  Showmessage('New learner '+ sexamnumber+' - successfully added');
  Sort;
  richedit1.clear;
  for k := 1 to icount do
    richedit1.lines.add(arrexamnumbers[k]);
  end;
```



```
procedure TForm1.Button4Click(Sender: TObject);
var
  myfile : textfile;
  sdelete : string;
  k, idelete : integer;
begin
  Assignfile(myfile, 'examnumbers.txt');
  Rewrite(myfile);
  sdelete := inputbox('','','');
  for k := 1 to icount do
   begin
    if arrexamnumbers[k] = sdelete then
      idelete := k;
   end;

   for k := idelete to icount - 1 do
     arrexamnumbers[k] := arrexamnumbers[k+1];

   dec(icount);
   arrexamnumbers[icount] := '';
   Showmessage('Learner deleted');

   Sort;

   for k := 1 to icount do
    begin
     writeln(myfile,arrexamnumbers[k]);
     richedit1.Lines.add(arrexamnumbers[k]);
    end;

   Closefile(myfile);
end;

end.
```