



Province of the
EASTERN CAPE
EDUCATION

**NATIONAL
SENIOR CERTIFICATE**

GRADE 12

SEPTEMBER 2011

**INFORMATION TECHNOLOGY P1
MEMORANDUM**

MARKS: 120

TIME: 3 hours

This memorandum consists of 12 pages.

QUESTION 1
(LO4 AS 11, 12)

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  adoquery1.active := false;
  adoquery1.SQL.Text := 'select *✓ from Employees✓ order by Surname, Initial✓';
  adoquery1.Active := true;
  SetGridColumnWidths(DBGrid1);
end;
```

(3)

```
procedure TForm1.Button2Click(Sender: TObject);
begin
  adoquery1.active := false;
  adoquery1.SQL.Text := 'select EmpNo, Surname, Initial from Employees✓ where  
Town = "East London"✓ and DriversLicence = Yes✓';
  adoquery1.Active := true;
  SetGridColumnWidths(DBGrid1);
end;
```

(3)

```
procedure TForm1.Button3Click(Sender: TObject);
begin
  adoquery1.active := false;
  adoquery1.SQL.Text := 'select EmpNo, Surname, DOB, Town✓ from  
Employees✓ where Surname like "Ma%"✓';
  adoquery1.Active := true;
  SetGridColumnWidths(DBGrid1);
end;
```

(3)

```
procedure TForm1.Button4Click(Sender: TObject);
begin
  adoquery1.active := false;
  adoquery1.SQL.Text := 'select Employees.EmpNo, Surname, Town,  
HouseholdNo✓ from Employees✓, Survey✓ where Employees.EmpNo =  
Survey.Empno✓ and Income = "MIDDLE"✓';
  adoquery1.Active := true;
  SetGridColumnWidths(DBGrid1);
end;
```

(5)

```
procedure TForm1.Button5Click(Sender: TObject);
var
  stown : string;
begin
  stown := inputbox('Enter Town', "", ""); ✓
  adoquery1.active := false;
  adoquery1.SQL.Text := 'select HouseholdNo, Employees.EmpNo, Surname✓
from Employees✓, Survey✓ where Employees.EmpNo = Survey.Empno ✓and
Town = "' + stown + "'"✓';
  adoquery1.Active := true;
  SetGridColumnWidths(DBGrid1);
end; (6)
```

```
procedure TForm1.Button6Click(Sender: TObject);
begin
  adoquery1.active := false;
  adoquery1.SQL.Text := 'select *✓ from Employees ✓where Title = "Mr"✓ and
Year(DOB) <= 1976✓';
  adoquery1.Active := true;
  SetGridColumnWidths(DBGrid1);
end; (4)
```

```
procedure TForm1.Button8Click(Sender: TObject);
var
  sempno : string;
begin
  sempno := inputbox('Enter Employee Number', "", ""); ✓
  adoquery1.active := false;
  adoquery1.SQL.Text := 'select count(*)✓ as [Households Counted] ✓from
Survey ✓where EmpNo = ' + sempno + "'"✓';
  adoquery1.Active := true;
  SetGridColumnWidths(DBGrid1);
end; (5)
```

```
procedure TForm1.Button9Click(Sender: TObject);
begin
  adoquery1.active := false;
  adoquery1.SQL.Text := 'select HouseholdNo, Dependants, Income✓,
Format(dependants * 260✓, "Currency"✓) as [State Contribution] ✓ from Survey
✓where Income = "MIDDLE"✓';
  adoquery1.Active := true;
  SetGridColumnWidths(DBGrid1);
end; (6)

end. [40]
```

QUESTION2

(LO4 AS 4)

2.1	2.1.1	Define a class	(1)
		private	(1)
		Declaring frace, fmale, ffemale, ftotal	(2)
	2.1.2	Constructor heading	(1)
		Assigning values to fields	(2)
		Initialising total	(1)
	2.1.3	Function toString heading	(1)
		Putting fields together	(1)
		#9 for columns	(1)
	2.1.4	Procedure calctotalpop heading	(1)
		Calculation – adding male and female	(1)
	2.1.5	Get total population heading	(1)
		Total assigned to result/function	(1)
	2.1.6	Get Race groups heading	(1)
Racel assigned to result/function		(1)	
2.1.7	Get number of males heading	(1)	
	Number of males assigned to result/function	(1)	
			[19]
2.2	2.2.1	Declare arrpop (global)	(1)
		Initialise counter	(1)
		Check if file exists	(1)
		Assignfile	(1)
		Reset	(1)
		While not eof() do	(1)
		readln	(1)
		Increase counter	(1)
		Get race	(1)
		Get male	(1)
		Get female	(1)
		Assign to arrpop	(2)
	2.2.2	Loop	(1)
		Call toString	(1)

2.2.3	Initialise total variable	(1)
	Loop	(1)
	Call calctotalpop	(1)
	Get totalpop and add to total	(1)
	Display total	(1)
2.2.4	Loop	(1)
	If getrace = white	(1)
	Call calctotalpop	(1)
	display	(1)
2.2.5	Initialise variables to store total population and total males	(2)
	loop	(1)
	Call Gettotalpop and add to variable	(1)
	Call Gettotalmales and add to variable	(1)
	Calculate percentage 2 decimals	(1)
	display	(1)

[31]**POSSIBLE SOLUTION**

unit Population;

interface

uses

 SysUtils;

type

 TPop = class

 private

 frace : string;

 fmale : integer;

 ffemale : integer;

 ftotal : integer;

 public

 constructor create(srace:string; imale, ifemale:integer);

 function GetTotalPop : integer;

 procedure CalcTotalPop;

 function toString : string;

 function GetRace : string;

 function GetMales : integer;

 end;

implementation

```
constructor TPop.create(srace:string; imale, ifemale:integer);
begin
    frace := srace;
    fmale := imale;
    ffemale := ifemale;
    ftotal := 0;
end;

procedure TPop.CalcTotalPop;
begin
    ftotal := fmale + ffemale;
end;

function TPop.GetTotalPop;
begin
    result := ftotal;
end;

function TPop.tostring : string;
begin
    result := frace + #9 + inttostr(fmale) + #9 + inttostr(ffemale);
end;

function TPop.GetRace : string;
begin
    result := frace;
end;

function TPop.GetMales : integer;
begin
    result := fmale;
end;

end.

unit Question2Memo_u;

interface

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

type

```
TForm1 = class(TForm)
  MainMenu1: TMainMenu;
  Population1: TMenuItem;
  Quite1: TMenuItem;
  PercentageofMales1: TMenuItem;
  TotalWhites1: TMenuItem;
  RichEdit1: TRichEdit;
  TotalPopulation1: TMenuItem;
  DisplayPopulation2001Stats1: TMenuItem;
  procedure FormCreate(Sender: TObject);
  procedure TotalPopulation1Click(Sender: TObject);
  procedure DisplayPopulation2001Stats1Click(Sender: TObject);
  procedure TotalWhites1Click(Sender: TObject);
  procedure PercentageofMales1Click(Sender: TObject);
  procedure Quite1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
```

var

```
Form1: TForm1;
arrpop : array[1..4] of TPop;
itotal : integer;
icount : integer;
implementation
```

{ \$R *.dfm }

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

var

```
myfile : textfile;
soneline : string;
ipos, imale, ifemale : integer;
srace : string;
```

begin

```
icount := 0;
if fileexists('Population.txt') <> true then
begin
  ShowMessage('File does not exist');
  Exit;
end;
```

```
Assignfile(myfile, 'Population.txt');
Reset(myfile);
while not eof(myfile) do
begin
  inc(icount);
  readln(myfile, soneline);
  ipos := pos(',',soneline);
  srace := copy(soneline,1,ipos-1);
  delete(soneline,1,ipos);
  ipos := pos(',',soneline);
  imale := strtoint(copy(soneline,1,ipos-1));
  delete(soneline,1,ipos);
  ifemale := strtoint(soneline);
  arrpop[icount] := Tpop.Create(srace,imale,ifemale);
end;
closefile(myfile);
end;
```

```
procedure TForm1.TotalPopulation1Click(Sender: TObject);
var
  k : integer;
begin
  richedit1.Clear;
  itotal := 0;
  for k := 1 to icount do
  begin
    arrpop[k].calctotalpop;
    itotal := itotal + (arrpop[k].gettotalpop);
  end;
  richedit1.Lines.add('Total Population: '+inttostr(itotal));
end;
```

```
procedure TForm1.DisplayPopulation2001Stats1Click(Sender: TObject);
var
  k : integer;
begin
  richedit1.Clear;
  richedit1.paragraph.tabcount := 3;
  richedit1.paragraph.tab[0] := 50;
  richedit1.paragraph.tab[1] := 100;
  richedit1.paragraph.tab[2] := 150;
  for k := 1 to icount do
    richedit1.lines.add(arrpop[k].tostring);
  end;
```



```

procedure TForm1.TotalWhites1Click(Sender: TObject);
var
  k : integer;
begin
  richedit1.Clear;
  for k := 1 to icount do
  begin
    if arrpop[k].getrace = 'White' then
    begin
      arrpop[k].calctotalpop;
      richedit1.Lines.add('Total Whites: '+floattostr(arrpop[k].gettotalpop));
    end;
  end;
end;

```

```

procedure TForm1.PercentageofMales1Click(Sender: TObject);
var
  itotalmales, itotalpop, k : integer;
begin
  richedit1.Clear;
  itotalmales := 0;
  itotalpop := 0;
  for k := 1 to icount do
  begin
    itotalpop := itotalpop + arrpop[k].gettotalpop;
    itotalmales := itotalmales + arrpop[k].getmales;
  end;
  richedit1.Lines.add('Percentage Males:
'+floattostrf(+itotalmales/itotalpop*100,ffixed,6,2)+'%');
end;

```

```

procedure TForm1.Quite1Click(Sender: TObject);
begin
  close;
end;

```

end.

[31]

QUESTION3

3.1	Procedure CreateEmpNum heading	(1)
	Passing of value parameter (counter)	(1)
	Passing of reference parameters (surname, initial)	(1)
	Passing of reference parameters (arrnew)	(1)
	Find Initial	(1)
	Find Surname	(1)
	Get first 3 letters of surname	(1)
	Get random number between 1000 and 1999	(2)
	Assemble new employee number and assign to arrnew	(2)
	Type declaration for arrnew	(2)

3.2	Procedure Save declared under public	(1)
	Assignfile	(1)
	Check if file exists; if not rewrite else append	(2)
	For loop	(1)
	Call createempnum procedure	(1)
	Create soneline	(1)
	Write to textfile	(1)
	Display message that text file has been created	(1)
3.3	Set up columns in richedit	(2)
	Heading in richedit	(1)
	For loop	(1)
	Call statement and parameter passing	(2)
	Display in richedit	(1)
	Call statement for the Save procedure	(1)
		[30]

POSSIBLE SOLUTION

unit Question3memo_u;

interface

uses

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

type

```
TForm1 = class(TForm)
  RichEdit1: TRichEdit;
  Button1: TButton;
  Button2: TButton;
  BitBtn1: TBitBtn;
  procedure FormCreate(Sender: TObject);
  procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  procedure SaveEmpNum;
end;
```

type

```
TarrNew = array[1..50] of string;
```

var

Form1: TForm1;
icount : integer;
arrempp : array[1..50] of string;

implementation

{ \$R *.dfm }

procedure TForm1.FormCreate(Sender: TObject);

var

myfile : textfile;
soneline : string;

begin

icount := 0;
if fileexists('EmpNumbers.txt') <> true then
begin
ShowMessage('File does not exist');
Exit;
end;

Assignfile(myfile, 'EmpNumbers.txt');

Reset(myfile);

while not eof(myfile) do

begin
readln(myfile, soneline);
inc(icount);
arrEmp[icount] := soneline;
end;

end;

procedure CreateEmpNum(k: integer; var ssurname, sinitia : string; var arrNew :
TarrNew);

var

iran, ipos : integer;
ssur : string;

begin

ipos := pos(',', arrempp[k]);
ssurname := copy(arrEmp[k], 1, ipos - 1);
ssur := copy(arrEmp[k], 1, 3);
sinitia := copy(arrEmp[k], pos(',', arrempp[k]) + 1, 1);
iran := random(1000) + 1000;
arrnew[k] := sinitia + ssur + inttostr(iran);

end;

```
procedure TForm1.SaveEmpNum;
var
  k : integer;
  myfile : textfile;
  soneline,ssurname,sinitial : string;
  arrnew : Tarrnew;
begin
  Assignfile(myfile, 'EmpNewNumbers.txt');
  if fileexists('EmpNewNumbers.txt') <> true then
    Rewrite(myfile)
  else
    Append(myfile);
  for k := 1 to icount do
    begin
      CreateEmpNum(k,ssurname, sinitial,arrnew);
      soneline := ssurname+', '+sinitial+', '+arrnew[k];
      writeln(myfile,soneline);
    end;
  ShowMessage('File successfully updated');
end;

procedure TForm1.Button1Click(Sender: TObject);
var
  k : integer;
  ssurname, sinitial : string;
  arrnew : Tarrnew;
begin
  richedit1.Paragraph.TabCount := 3;
  richedit1.Paragraph.tab[0] := 30;
  richedit1.Paragraph.tab[1] := 50;
  richedit1.Paragraph.tab[2] := 70;
  richedit1.lines.add('SURNAME'+#9+'INITIAL'+#9+'EMPLOYEE NUMBER');
  for k := 1 to icount do
    begin
      CreateEmpNum(k, ssurname, sinitial,arrnew);
      richedit1.Lines.add(ssurname+#9+sinitial+#9+arrNew[k]);
    end;
  SaveEmpNum;
end;

end.
```

[30]

TOTAL: 120