Province of the
EASTERN CAPE
EDUCATION

# NATIONAL SENIOR CERTIFICATE

# GRADE 11

# NOVEMBER 2016

# INFORMATION TECHNOLOGY P1 MEMORANDUM

**MARKS: 150**

This memorandum consists of 15 pages.

## GENERAL PROGRAMMING SKILLS

| QUESTION 1 | | MAXIMUM MARKS | MARKS ACHIEVED |
|---|---|---|---|
| 1.1 | Get name ✓<br>Get date of birth✓<br>Extract the year from date of birth ✓ and calculate the age ✓<br>If age <= 18 ✓<br>Display name and Junior category ✓<br>Else✓<br>Display name and Senior category ✓ | **8** | |
| 1.2 | If both checkboxes ✓ are selected then discount is 16% ✓<br>If club checkbox is selected then discount is 6%✓<br>If CSA checkbox is selected then discount is 10% ✓<br>If no checkboxes are selected then no Discount ✓<br>Display message ✓ | **6** | |
| 1.3 | Initialise the code variable ✓<br>Get the text from combobox ✓<br>Get the first letter of every word to create the code (use a loop) ✓✓✓✓✓<br>Randomise a number between 1000 and 4999 ✓<br>Display race number ✓ | **9** | |
| 1.4 | Use either a case statement or nested if statement to determine the provincial fees ✓✓✓✓<br>If the domestic racing licence checkbox is selected✓ then 350 must be added ✓ else 0 must be added ✓<br>If the CSA checkbox is selected ✓then 0 must be added ✓else 35 must be added✓<br>Add all amounts together (150+provincial+racing+CSA) ✓<br>Display the total ✓ as currency rounded to 2 decimal places ✓ | **13** | |
| | | **36** | |

| QUESTION 2 | | MAXIMUM MARKS | MARKS ACHIEVED |
|---|---|---|---|
| 2.1 | Declare arrname and arraverage ([1..50], correct data types) ✓✓<br>Declare global counter✓<br><br>**FormCreate:**<br>Declare local variables✓<br>Test if file exists✓<br>Showmessage✓<br>Exit✓<br>Assignfile✓<br>Reset ✓<br>Initialise the counter✓<br>Loop through text file✓<br>Read line from text file✓<br>Increase counter✓<br>Extract the surname✓<br>Extract the name✓<br>Assign the name and surname to the array✓<br>Extract the first time✓<br>Extract the second time✓<br>Calculate the average time✓ and assign to array✓<br>Closefile✓ | **21** | |
| 2.2 | Loop through array – correct counter✓<br>Display arrname✓<br>Display total number of cyclists✓ on next line✓ with a label✓ | **5** | |
| 2.3 | Outer loop✓<br>Inner loop✓<br>If statement✓<br>Code to swap the average✓✓✓<br>Code to swap the name✓✓<br>Assignfile✓<br>Rewrite✓<br>Loop to get top three✓<br>Compile line – counter✓+name✓+average✓ (2 decimal places) ✓<br>Write to text file✓<br>Display in richedit✓<br>Display a message that text file was written✓<br>Closefile ✓ | **19** | |
| | | **45** | |

| QUESTION 3 | | MAXIMUM MARKS | MARKS ACHIEVED |
|---|---|---|---|
| 3.1 | Function Age; ✓ id as parameter;✓ integer return type✓<br>Get the YY from the parameter✓<br>Determine whether ‘19’/1900✓✓ or ‘20’/2000 ✓✓ must be added to the year<br>Calculate the age✓ | **9** | |
| 3.2 | Procedure Gender; ✓id as parameter (value parameter); ✓gender must be returned (variable parameter) ✓<br>Get 7th digit of ID✓ and check if less than 4✓<br>Then assign ‘female’✓<br>else✓<br>assign ‘male’✓ | **8** | |
| 3.3 | Procedure Racenumber✓category as parameter✓<br>Randomise a number between 1000 and 9999✓<br>Compile the race number: Get first three characters of the category✓ and uppercase it✓ and add the random number✓<br>Display the race number✓ | **7** | |
| 3.4 | Get the name✓<br>Get the ID✓<br>Call statement for Gender✓<br>Calculate the categories calling the Age method:<br>If age >= 13 and age <=18✓ – Junior ✓<br>If age >= 19 and age <=39✓ – Elite ✓<br>If age >= 40 and age <= 59✓ – Veteran ✓<br>If age >= 60 – Masters✓<br>Display the name and Gender on two lines ✓<br>Call statement for Racenumber ✓ | **12** | |
| | | **36** | |

| QUESTION 4 | | MAXIMUM MARKS | MARKS ACHIEVED |
|---|---|---|---|
| 4.1 | **NO FILTERS ALLOWED**<br>Open the table✓<br>Go to first record✓<br>Initialise counter✓<br>Display heading✓<br>Conditional loop✓<br>If race is in PE✓<br>Display date✓ and race✓<br>Increase counter✓<br>Next record✓<br>Display total races in PE✓<br>Close table✓ | **12** | |
| 4.2 | Conditional loop✓<br>If type is 'ROAD' then increase counter✓<br>If type is 'MTB' then increase counter✓<br>Next record✓<br>Display total road races✓<br>Display total mtb races✓<br><br>**ALTERNATIVE (using filters):**<br>Filter for type = ROAD✓<br>Record Count✓<br>Display total road races✓<br>Filter for type = MTB✓<br>Record Count✓<br>Display total mtb races✓ | **6** | |
| 4.3 | Get input✓ and change to uppercase✓<br>If race is in table (locate) ✓<br>Display race and venue✓ and that it is found✓<br>else✓<br>Display not found✓ | **7** | |
| 4.4 | Find correct record (Cape Argus) ✓<br>Edit✓<br>Add email address to contact field✓<br>Post✓ | **4** | |
| 4.5 | Find the correct record (Cabbage Patch) ✓✓<br>Delete✓<br>Display a message that the race has been deleted from the table✓ | **4** | |
| | | **33** | |

## SAMPLE SOLUTIONS

### Question 1

```
unit Question1_u;

interface

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

type
  TForm1 = class(TForm)
    Panel1: TPanel;
    Edit1: TEdit;
    Label1: TLabel;
    Edit2: TEdit;
    Label2: TLabel;
    Button1: TButton;
    Edit3: TEdit;
    Panel2: TPanel;
    CheckBox1: TCheckBox;
    Button2: TButton;
    CheckBox2: TCheckBox;
    Label3: TLabel;
    Panel3: TPanel;
    ComboBox1: TComboBox;
    Label4: TLabel;
    Button3: TButton;
    Label5: TLabel;
    Panel4: TPanel;
    Button4: TButton;
    Label7: TLabel;
    RadioGroup1: TRadioGroup;
    CheckBox3: TCheckBox;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}
```

```
procedure TForm1.Button1Click(Sender: TObject);
var
  sname, sdob : string;
  iage : integer;
begin
  sname := edit1.text;
  sdob := edit2.text;
  iage := 2016 - strtoint(copy(sdob,1,4));
  if iage <= 18 then
    edit3.text := sname + ': Junior Category'
  else
    edit3.text := sname + ': Senior Category';
end;

procedure TForm1.Button2Click(Sender: TObject);
var
  sdiscount : string;
begin
 if checkbox1.checked and checkbox2.checked then
          sdiscount := '16%'
        else
  if checkbox1.Checked then
    sdiscount := '6%'
  else
    if checkbox2.checked then
      sdiscount := '10%'
       else
          sdiscount := '0%';
  label3.caption := sdiscount + ' discount on race fees';
end;

procedure TForm1.Button3Click(Sender: TObject);
var
  srace, scode : string;
  ipos, iran : integer;
begin
  randomize;
  scode := ' ';
  srace := combobox1.text;
  ipos := pos(' ',srace);
  scode := scode + srace[1];
  while (ipos <> 0) do
   begin
    delete(srace,1,ipos);
    scode := scode + srace[1];
    ipos := pos(' ',srace);
  end;
  iran := randomrange(1000,5000);
  label5.caption := 'Race Number is: ' + scode + inttostr(iran);
end;
```

```
procedure TForm1.Button4Click(Sender: TObject);
var
  rprov, rrace, rday, rtotal : real;
begin
  case radiogroup1.ItemIndex of
    0,3,5: rprov := 75;
    1,2,4,6,9: rprov := 50;
    7,8: rprov := 25;
  end;

  if checkbox3.checked then
    rrace := 350
  else
    rrace := 0;

  if checkbox2.checked then
    rday := 0
  else
    rday := 35;

  rtotal := 150 + rprov + rrace + rday;
  Showmessage('Total to be paid to CSA: '+floattostrf(rtotal,ffcurrency,10,2));
end;

end.
```

## Question 2

```
unit Question2_u;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, ExtCtrls;

type
  TForm1 = class(TForm)
    Button1: TButton;
    redoutput: TRichEdit;
    Button2: TButton;
    Panel1: TPanel;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
   Form1: TForm1;
  arrname : array[1..50] of string;
  arraverage : array[1..50] of real;
  icount : integer;
implementation

{$R *.dfm}
procedure TForm1.FormCreate(Sender: TObject);
var
  ipos:integer;
  myfile : textfile;
  rfirst, rsecond : real;
  soneline,sname,ssurname : string;
begin
  if fileexists('timetrial.txt') <> true then
  begin
    ShowMessage('File does not exist');
    Exit;
  end;
  Assignfile(myfile,'timetrial.txt');
  Reset(myfile);
  icount := 0;
  while not eof(myfile) do
  begin
    readln(myfile,soneline);
    inc(icount);
    ipos := pos('#',soneline);
    ssurname := copy(soneline,1,ipos-1);
    delete(soneline,1,ipos);
    ipos := pos('#',soneline);
    sname := copy(soneline,1,ipos-1);
    delete(soneline,1,ipos);
    arrname[icount] := sname + ' ' + ssurname;
    ipos := pos('#',soneline);
    rfirst := strtofloat(copy(soneline,1,ipos-1));
    delete(soneline,1,ipos);
    rsecond := strtofloat(soneline);
    arraverage[icount] := (rfirst + rsecond)/2;
end;
closefile(myfile);
end;

procedure TForm1.Button1Click(Sender: TObject);
var
  k : integer;
begin
  for k := 1 to icount do
    redoutput.lines.add(arrname[k]);
  redoutput.lines.add(#13+'Total Number of Cyclists: '+inttostr(icount));
end;

procedure TForm1.Button2Click(Sender: TObject);
var
  k,l : integer;
  myfile : textfile;
  stemp,soneline : string;
  rtemp : real;
begin
  for k := 1 to icount - 1 do
    for l := k + 1 to icount do
      if arraverage[k] > arraverage[l] then
        begin
          rtemp := arraverage[k];
          arraverage[k] := arraverage[l];
          arraverage[l] := rtemp;
          stemp := arrname[k];
          arrname[k] := arrname[l];
          arrname[l] := stemp;
        end;
  Assignfile(myfile,'top3.txt');
  Rewrite(myfile);
  for k := 1 to 3 do
    begin
      soneline :=inttostr(k)+#9+arrname[k] + #9 + floattostrf(arraverage[k],fffixed,10,2);
      writeln(myfile,soneline);
      redoutput.lines.add(soneline);
    end;
  Showmessage('File successfully written');
  closefile(myfile);
end;
end.
```

## Question 3

```
unit question3_u;

interface

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

type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Label1: TLabel;
    Button1: TButton;
    Edit2: TEdit;
    Label2: TLabel;
    RichEdit1: TRichEdit;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }

    procedure Gender(sid :string; var sgender : string);
    procedure RaceNumber(scategory : string);
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

function Age(sid : string) : integer;
var
  iyy, iyear : integer;
begin
  iyy := strtoint(copy(sid,1,2));
  if iyy > 16 then
    iyear := 1900 + iyy
  else
    iyear := 2000 + iyy;
  age := currentyear() - iyear;
end;

procedure TForm1.RaceNumber(scategory : string);
var
  iran : integer;
  sracenumber : string;
```

```
begin
  iran := randomrange(1000,9999);
  sracenumber := uppercase(copy(scategory,1,3)) + inttostr(iran);
  richedit1.lines.add('Race Number: '+sracenumber);
end;

procedure TForm1.Button1Click(Sender: TObject);
var
  sname,sid, sgender, scategory : string;
begin
  sname := edit1.text;
  sid := edit2.text;
  Gender(sid,sgender);
  if (Age(sid) >= 13) and (Age(sid) <= 18) then
    scategory := 'Junior'
  else
    if (Age(sid) >= 19) and (Age(sid) <= 39) then
      scategory := 'Elite'
    else
      if (Age(sid) >= 40) and (Age(sid) <= 59) then
        scategory := 'Veteran'
      else
        scategory := 'Masters';
    richedit1.lines.add('Name: '+sname + #13 + 'Gender: '+sgender);
    RaceNumber(scategory);
end;

procedure Tform1.Gender(sid :string; var sgender : string);
begin
   if strtoint(copy(sid,7,1)) < 4 then
     sgender := 'Female'
   else
     sgender := 'Male';
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
  randomize;
end;

end.
```

```
unit Question4_u;

interface

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

type
  TForm1 = class(TForm)
    ADOTable1: TADOTable;
    DataSource1: TDataSource;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    RichEdit1: TRichEdit;
    BitBtn1: TBitBtn;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.BtnQ4_1Click(Sender: TObject);
var
  iracepe : integer;
begin
  Adotable1.Open;
  adotable1.First;
  iracepe := 0;
  richedit1.Lines.Add('Date'+#9+'Race');
  while NOT adotable1.eof do
  begin
    if adotable1['Where'] = 'PE' then
    begin
      richedit1.Lines.Add(datetostr(adotable1['Date'])+#9+adotable1['Race']);
      inc(iracepe);
    end;
  adotable1.Next;
  end;
  richedit1.Lines.Add('Total Races in PE: '+inttostr(iracepe));
  adotable1.Close;
```

```
end;

procedure TForm1.BtnQ4_2Click(Sender: TObject);
var
  iroad, imtb : integer;
begin
  richedit1.Clear;
  adotable1.Open;
  adotable1.First;
  iroad := 0;
  imtb := 0;
  while NOT adotable1.eof do
  begin
    if adotable1['Type'] = 'ROAD' then
      inc(iroad)
    else
      if adotable1['Type'] = 'MTB' then
        inc(imtb);
  adotable1.Next;
  end;
  richedit1.Lines.Add('Road Races: '+inttostr(iroad));
  richedit1.Lines.Add('MTB Races: '+inttostr(imtb));
  adotable1.Close;
end;

procedure TForm1.BtnQ4_3Click(Sender: TObject);
var
  sinput : string;
begin
  richedit1.Clear;
  sinput := uppercase(inputbox('Enter Race','','Bay to Beach'));
  adotable1.Open;
  adotable1.First;
  if adotable1.Locate('Race',sinput,[lopartialkey]) then
    richedit1.Lines.Add(adotable1['Race']+#9+adotable1['Where']+' - found')
  else
    richedit1.Lines.Add('Race not found');
  adotable1.Close;
end;

procedure TForm1.btnQ4_5Click(Sender: TObject);
begin
  adotable1.Open;
  adotable1.First;
  while not adotable1.eof do
  begin
    if adotable1['Resies'] = 'THE CABBAGE PATCH' then
     begin
      adotable1.Delete;
      showmessage('Race has been cancelled');
     end;
    adotable1.Next;
  end;
```

```
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  richedit1.Paragraph.Tabcount := 1;
  richedit1.Paragraph.Tab[0] := 75;
 adotable1.Open;
  adotable1.First;
  while not adotable1.eof do
  begin
    if adotable1['Race’] = 'CAPE ARGUS CYCLE TOUR' then
     begin
      adotable1.Edit;
        adotable1['Contact'] := 'argus@capeargus.co.za';
     end;
    adotable1.Next;
  end;

end;

end.
```