**Question1\_memo; (30 marks)**

**(LO 4 AS 11, 12)**

procedure TForm1.Button1Click(Sender: TObject);

begin

qrySchool.active := false;

qrySchool.SQL.Text := 'Select \*✓ from EducatorsTb ✓order by Surname✓';

qrySchool.Active := true;

SetGridColumnWidths(DBGrid1);

end;

procedure TForm1.Button2Click(Sender: TObject);

begin

qrySchool.active := false;

qrySchool.SQL.Text := 'Select Title, Surname, TeacherCode✓ from EducatorsTb✓ where (subjects = "ENG"✓ or subjects = "AFR"✓ or subjects = "XHO"✓) ';

qrySchool.Active := true; OR where (subjects IN [“ENG”,”AFR”,”XHO”)

SetGridColumnWidths(DBGrid1);

end;

procedure TForm1.Button3Click(Sender: TObject);

var

ssubject : string;

begin

ssubject := inputbox('Enter Subject','',''); ✓

qrySchool.active := false;

qrySchool.SQL.Text := 'Select count(\*)✓as [Potential Subject Numbers] ✓ from LearnersTb✓ where (S5 = "'+ssubject+'")'; ✓

qrySchool.Active := true;

SetGridColumnWidths(DBGrid1);

end;

procedure TForm1.Button5Click(Sender: TObject);

begin

qrySchool.active := false;

qrySchool.SQL.Text := 'Select Name, Surname✓ from LearnersTb ✓where Teachers like "%AM%"'; ✓

qrySchool.Active := true;

SetGridColumnWidths(DBGrid1);

end;

procedure TForm1.Button7Click(Sender: TObject);

begin

qrySchool.active := false;

qrySchool.SQL.Text := 'Update LearnersTb✓ Set S5 = "LSC"✓ where S5 = "ART"'; ✓

qrySchool.Active := true;

qrySchool.SQL.Text := 'Select \*✓ from LearnersTb'; ✓

SetGridColumnWidths(DBGrid1);

end;

procedure TForm1.Button8Click(Sender: TObject);

begin

qrySchool.active := false;

qrySchool.SQL.Text := 'Select class, count(class) ✓ AS [Class Sizes] ✓ from LearnersTb✓ group by Class✓';

qrySchool.Active := true;

SetGridColumnWidths(DBGrid1);

end;

procedure TForm1.Button9Click(Sender: TObject);

begin

qrySchool.active := false;

qrySchool.SQL.Text := 'Select \*✓ from LearnersTb ✓where (S7 = "CONS" ✓or S7 = "TOUR"✓) and (class = "G"✓)';

qrySchool.Active := true;

SetGridColumnWidths(DBGrid1);

end;

**Question2\_memo; (55 marks)**

**(LO 4 AS 4)**

|  |  |  |
| --- | --- | --- |
| 2.1.1 | Define a class | 1 |
|  | private | 1 |
|  | Declaring fname, feng, fmaths, faverage | 2 |
| 2.1.2 | Constructor heading | 1 |
|  | Assigning values to fields | 3 |
|  | Initialising faverage | 1 |
| 2.1.3 | Procedure CalcAverage heading | 1 |
|  | calculation | 2 |
| 2.1.4 | Function Qualify – Boolean | 1 |
|  | If average >= 60 | 1 |
|  | Qualify true else qualify false | 2 |
| 2.1.5 | Function tostring heading | 1 |
|  | Putting fields together | 1 |
|  | #9 for columns | 1 |
| 2.1.6 | GetName heading | 1 |
|  | fname assigned to result/function | 1 |
| 2.1.7 | GetAverage heading | 1 |
|  | faverage assigned to result/function | 1 |
|  |  | [23] |
|  |  |  |
| 2.2.1 | Declare arrlearners (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 name | 1 |
|  | Get eng mark | 1 |
|  | Get maths mark | 1 |
|  | Assign to arrlearner | 2 |
| 2.2.2 | Loop | 1 |
|  | Call CalcAverage | 1 |
|  | Call getname and getaverage 🡪 display | 2 |
| 2.2.3 | Loop | 1 |
|  | Call CalcAverage | 1 |
|  | Call qualify | 1 |
|  | If qualify = true display tostring | 1 |
| 2.2.4 | Inputbox | 1 |
|  | Set flag variable to false | 1 |
|  | Loop (counter less than number of learners and flag variable false) | 2 |
|  | If getname = input from inputbox | 2 |
|  | Then flag variable = true | 1 |
|  | Else flag variable = false | 1 |
|  | Increase counter | 1 |
|  | If flag variable = true then display that name was found | 2 |
|  |  | [32] |

**POSSIBLE SOLUTION**

**unit Subjects\_u;**

interface

uses

SysUtils;

type

TLearner = class

private

fname : string;

feng : integer;

fmaths : integer;

faverage : real;

public

constructor create (sname : string; ieng, imaths : integer);

procedure CalcAverage;

function tostring : string;

function getname : string;

function getaverage : real;

function qualify : boolean;

end;

implementation

constructor TLearner.create (sname : string; ieng, imaths : integer);

begin

fname := sname;

feng := ieng;

fmaths := imaths;

faverage := 0;

end;

procedure TLearner.CalcAverage;

begin

faverage := (feng + fmaths)/2;

end;

function Tlearner.tostring : string;

begin

result := fname + #9 + inttostr(feng) + #9 + inttostr(fmaths) + #9 + floattostrf(faverage,fffixed,3,1);

end;

function TLearner.getname : string;

begin

result := fname;

end;

function TLearner.getaverage : real;

begin

result := faverage;

end;

function TLearner.qualify : boolean;

begin

if faverage >= 60 then

qualify := true

else

qualify := false;

end;

end.

**unit Question2\_u;**

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, Menus, StdCtrls, ComCtrls, Subjects\_u;

type

TForm1 = class(TForm)

MainMenu1: TMainMenu;

RichEdit1: TRichEdit;

Options1: TMenuItem;

Average1: TMenuItem;

ITPhysicalScienceMaths1: TMenuItem;

SearchforaLearner1: TMenuItem;

Exit1: TMenuItem;

procedure FormActivate(Sender: TObject);

procedure Average1Click(Sender: TObject);

procedure ITPhysicalScienceMaths1Click(Sender: TObject);

procedure SearchforaLearner1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

icount : integer;

arrLearners : array[1..50] of TLearner;

implementation

{$R \*.dfm}

procedure TForm1.FormActivate(Sender: TObject);

var

myfile : textfile;

soneline : string;

ipos, ieng, imaths : integer;

sname : string;

begin

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

begin

ShowMessage('File does not exist');

Exit;

end;

Assignfile(myfile, 'Learner.txt');

Reset(myfile);

icount := 0;

while not eof(myfile) do

begin

readln(myfile,soneline);

inc(icount);

ipos := pos(',',soneline);

sname := copy(soneline, 1, ipos - 1);

delete(soneline, 1, ipos);

ipos := pos(',',soneline);

ieng := strtoint(copy(soneline, 1, ipos - 1));

delete(soneline, 1, ipos);

imaths := strtoint(soneline);

arrLearners[icount] := TLearner.create(sname, ieng, imaths);

end;

closefile(myfile);

end;

procedure TForm1.Average1Click(Sender: TObject);

var

k : integer;

begin

richedit1.lines.add('Name' + #9 + #9 + 'Average');

for k := 1 to icount do

begin

arrLearners[k].CalcAverage;

richedit1.Lines.add(arrLearners[k].getname + #9 + floattostrf(arrLearners[k].getaverage,fffixed,6,0));

end;

end;

procedure TForm1.ITPhysicalScienceMaths1Click(Sender: TObject);

var

k : integer;

begin

richedit1.lines.add('Name' + #9 +#9+ 'English' + #9 + 'Maths' + #9 + 'Average');

for k := 1 to icount do

begin

arrLearners[k].CalcAverage;

if arrlearners[k].qualify then

richedit1.Lines.Add(arrlearners[k].tostring);

end;

end;

procedure TForm1.SearchforaLearner1Click(Sender: TObject);

var

sinput : string;

k : integer;

bfound : boolean;

begin

sinput := inputbox('Enter Name','','');

bfound := false;

k := 1;

while (k <= icount) and (bfound = false) do

begin

if arrLearners[k].getname = sinput then

bfound := true

else

bfound := false;

inc(k);

end;

if bfound = true then

ShowMessage(sinput + ' found.');

if bfound = false then

ShowMessage(sinput + ' not found.');

end;

end.

**Question3\_memo; (35 marks)**

|  |  |  |
| --- | --- | --- |
| 3.1 | For loop - columns | 1 |
|  | For loop – rows | 1 |
|  | Assign random values (100 – 600) to stringgrid | 2 |
|  | Assign column headings – using a for loop (2), alternative method (1) | 2 |
|  | Assign row headings – using a for loop (2), alternative method (1) | 2 |
|  |  | [8] |
|  |  |  |
| 3.2 | Procedure Calculate Totals | 1 |
|  | Heading for row in stringgrid | 1 |
|  | For loop – column | 1 |
|  | Initialise total variable | 1 |
|  | For loop – row | 1 |
|  | Calculate total of all values in each column | 1 |
|  | Display the total in the stringgrid | 1 |
|  |  | [7] |
|  |  |  |
| 3.3 | Procedure CalculateAvgWeek | 1 |
|  | Heading for row in stringgrid | 1 |
|  | For loop | 1 |
|  | Initialise total variable | 1 |
|  | Add all values in columns (for each class) | 2 |
|  | Divide totals by 7 | 1 |
|  | Display the average formatted to 2 decimals in the stringgrid | 2 |
|  |  | [9] |
|  |  |  |
| 3.4 | CalculateTotals call statement | 1 |
|  | CalculateAvgWeek call statement | 1 |
|  |  | [2] |
|  |  |  |
| 3.5 | Initialise high variable to 0 | 1 |
|  | For loop - row | 1 |
|  | Initialise sum variable | 1 |
|  | For loop - column | 1 |
|  | Add values for each class | 1 |
|  | If sum is higher than high | 1 |
|  | Assign sum’s value to high | 1 |
|  | Assign the row value to a variable | 1 |
|  | Display message of winning class | 1 |
|  |  | [9] |

**POSSIBLE SOLUTION**

unit question3\_u;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,

Dialogs, StdCtrls, Grids;

type

TForm1 = class(TForm)

StringGrid1: TStringGrid;

Button1: TButton;

Button2: TButton;

procedure FormCreate(Sender: TObject);

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

private

{ Private declarations }

publi

procedure CalculateTotals;

procedure CalculateAvgWeek;

end;

var

Form1: TForm1;

implementation

{$R \*.dfm}

procedure TForm1.FormCreate(Sender: TObject);

var

irow, icol : integer;

begin

randomize;

for icol := 1 to 4 do

for irow := 1 to 7 do

stringgrid1.Cells[icol,irow] := inttostr(random(500)+101);

for icol := 1 to 4 do

stringgrid1.Cells[icol,0] := 'Week '+inttostr(icol);

for irow := 1 to 7 do

stringgrid1.cells[0,irow] := 'Class '+inttostr(irow);

end;

procedure TForm1.CalculateTotals;

var

icol,irow, itotal : integer;

begin

stringgrid1.cells[0,8] := 'Totals';

for icol := 1 to 4 do

begin

itotal := 0;

for irow := 1 to 7 do

begin

itotal := itotal + strtoint(stringgrid1.cells[icol,irow]);

end;

stringgrid1.Cells[icol,irow] := inttostr(itotal)

end;

end;

procedure TForm1.CalculateAvgWeek;

var

irow, icol : integer;

rtotal, raverage : real;

begin

stringgrid1.cells[0,9] := 'Average';

for icol := 1 to 4 do

begin

rtotal := rtotal + strtofloat(stringgrid1.cells[icol,8]);

raverage := rtotal/7;

stringgrid1.cells[icol,9] := floattostrf(raverage,fffixed,6,2);

rtotal := 0;

end;

end;

procedure TForm1.Button1Click(Sender: TObject);

begin

CalculateTotals;

CalculateAvgWeek;

end;

procedure TForm1.Button2Click(Sender: TObject);

var

irow, ihigh, iclass, icol, isum,k,l,itemp : integer;

arrclass : array[1..7] of integer;

begin

ihigh := 0;

for irow := 1 to 7 do

begin

isum := 0;

for icol := 1 to 4 do

begin

isum := isum + strtoint(stringgrid1.cells[icol,irow]);

end;

if isum > ihigh then

begin

ihigh := isum;

iclass := irow;

end;

end;

ShowMessage('Winning class: '+inttostr(iclass));

end;

end.