

```
unit Question1_U;
```

```
  frmQuestion1: TfrmQuestion1;  
  sSlogan : String;
```

```
implementation
```

```
{ $\$R$  *.dfm}
```

```
// =====
```

```
// Question 1.1
```

```
// =====
```

```
procedure TfrmQuestion1.btnQ1_1_1Click(Sender: TObject);
```

```
begin
```

```
  ImgRecycle.Picture.LoadFromFile('recycle.jpg');
```

```
  ImgRecycle.Stretch:=true;
```

```
  pnlSlogan.Font.Style:=[fsBold];
```

```
  pnlSlogan.Font.Size:=20;
```

```
  pnlSlogan.Color:=clSkyBlue;
```

```
end;
```

```
// =====
```

```
// Question 1.2
```

```
// =====
```

```
procedure TfrmQuestion1.btnQ1_1_2Click(Sender: TObject);
```

```
var rArea : real;
```

```
  iHeight, iradius : integer;
```

```
begin
```

```
  iHeight:= strToInt(cmbQ1_1_2.Text);
```

```
  iRadius:= sedQ1_1_2.Value;
```

```
  rArea := 2*Pi*iradius*iradius+2*Pi*iradius*iHeight;
```

```
  edtQ1_1_2.Text:=floatToStrF(rArea,ffFixed,10,2);
```

```
end;
```

```
// =====
```

```
// Question 1.3
```

```
// =====
```

```
procedure TfrmQuestion1.btnQ1_1_3Click(Sender: TObject);
```

```
var iAngle1,iAngle2,iAngle3,iSum : integer;
```

```
begin
```

```
  //Given code
```

```
  iAngle1 := StrToInt(edtQ1_3_1.Text);
```

```
  iAngle2 := StrToInt(edtQ1_3_2.Text);
```

```
  iAngle3 := StrToInt(edtQ1_3_3.Text);
```

```
//1.3
redQ1_3.Clear;
isum := iAngle1+iAngle2+iAngle3;
if (isum<>180) then redQ1_3.Lines.add('Not a Triangle')
else
begin
if (iAngle1=iAngle2) AND (iAngle2=iAngle3) then
redQ1_3.Lines.Add('Equilateral Triangle')
Else
begin
if (iAngle1=iAngle2) OR (iAngle2=iAngle3) OR (iAngle1=iAngle3) then
redQ1_3.Lines.Add('Isoceles Triangle')
Else redQ1_3.Lines.Add('Scalene triangle');
end;
end;
end;

// =====
// Question 1.4
// =====
procedure TfrmQuestion1.btnQ1_1_4Click(Sender: TObject);
var iTerm : integer;
sLine:string;
begin
//1.4
iTerm := strToint(inputbox('', 'Enter starting value? ', '100'));
sLine:= intToStr(iTerm);
while iTerm<200 do
begin
iTerm:=iTerm+13;
sLine:=sLine+#9+intToStr(iTerm);
end;
RedQ1_4.Lines.Add(sLine);
end;

// =====
// Question 1.5
// =====
procedure TfrmQuestion1.btnQ1_1_5Click(Sender: TObject);
var
sLine, sWord : string;
i,j: Integer;
ctemp: Char;

begin
//Given Code
sLine:=edtQ1_5_1.Text;
```

```
//1.5
for i := 1 to length(sLine) do
begin
  if i mod 2 <> 0 then sWord:=sWord+sLine[i];
end;
for i := 1 to length(sWord) do
begin
  for j := 1 to length(sWord) do
begin
  if sWord[i]>sWord[j] then
begin
  cTemp:=sWord[i];
  sWord[i]:=sWord[j];
  sWord[j]:=cTemp;
end;
end;
end;
edtQ1_5_2.Text:=sWord;
end;

end.
```

unit Question2_U;

```
// =====  
// Question 2.1.1  
// =====
```

```
procedure TfrmQuestion2.btnQ2_1_1Click(Sender: TObject);  
var  
  sSQL1: String;  
begin  
  // Question 2.1.1
```

```
  sSQL1 := 'SELECT * FROM tblClients WHERE City LIKE "Durban" '+  
  'ORDER BY ClientSurname DESC';
```

```
  // Provided code - do not change  
  dbCONN.runSQL(sSQL1);  
end;
```

```
// =====  
// Question 2.1.2  
// =====
```

```
procedure TfrmQuestion2.btnQ2_1_2Click(Sender: TObject);  
var  
  sSQL2: String;  
begin  
  // Question 2.1.2
```

```
  sSQL2 := 'SELECT CollectionID, CollectionDate, NumberOfCans, KgsOfCardboard FROM '+  
  'tblCollection WHERE Month(CollectionDate) BETWEEN 1 AND 3 AND NOT(Paid)';
```

```
  // Provided code - do not change  
  dbCONN.runSQL(sSQL2);  
end;
```

```
// =====  
// Question 2.1.3  
// =====
```

```
procedure TfrmQuestion2.btnQ2_1_3Click(Sender: TObject);  
var  
  sSQL3, sLetter: String;  
begin  
  // Provided code
```

```
  sLetter:= InputBox('','Enter Letter to Search for?','');
```

```
// Question 2.1.3
```

```
sSQL3 := 'SELECT * FROM tblCollection WHERE numberOfCans > ' + sNumber +  
' and collectionDate > #2023/04/20#';
```

```
// Provided code - do not change  
dbCONN.runSQL(sSQL3);  
end;
```

```
// Provided code - do not change  
dbCONN.runSQL(sSQL3);  
end;
```

```
// =====  
// Question 2.1.4  
// =====
```

```
procedure TfrmQuestion2.btnQ2_1_4Click(Sender: TObject);  
var  
  sSQL4: String;  
begin  
  // Question 2.1.4
```

```
  sSQL4 := 'SELECT ClientName, FORMAT(SUM(KgsOfCardboard)/5*4,"currency") ' +  
  'AS [Income From Cardboard] FROM tblClients, tblCollection WHERE ' +  
  'tblClients.ClientID = tblCollection.ClientID and Paid GROUP BY ClientName';
```

```
// Provided code - do not change  
dbCONN.runSQL(sSQL4);  
end;
```

```
// =====  
// Question 2.1.5  
// =====
```

```
procedure TfrmQuestion2.btnQ2_1_5Click(Sender: TObject);  
var  
  sSQL5 : String;  
  bChange : boolean;  
begin  
  // Question 2.1.5
```

```
  sID := InputBox("", 'Enter clientID', "");  
  sSQL5 :=  
  'UPDATE tblCollection SET ElectronicPayment=true WHERE ' + 'ClientID= '+quotedstr(sID);
```

```
// Provided code - do not change  
dbCONN.ExecuteSQL(sSQL5, bChange);
```

```
if bChange then
begin
  MessageDlg('Database updated', mtInformation, [mbOK], 0);
end;
end;

// Delphi code section
// =====
// Question 2.2.1
// =====

procedure TfrmQuestion2.lblQuantityClick(Sender: TObject);
begin
// Question 2.2.1
tblClients.First;
tblClients['ClientID']:='CHA01';
tblClients['ClientName']:='Charles';
tblClients['ClientSurname']:='du Bolt';
tblClients['Address']:='24 Van Wouw Street';
tblClients['City']:='Cape Town';
tblClients.Post;
end;

// =====
// Question 2.2.2
// =====

procedure TfrmQuestion2.btnQ2_2_2Click(Sender: TObject);
var
  iSumClient, iSumCompany: integer;
  sClientsName, sClientID : String;
  iMonth : integer;
begin
// Provided code - do not change
redQ2_2_2.Clear;
iMonth := rgpQ2_2_2.ItemIndex+1;
// =====
// Question 2.2.2
iSumClient:=0;
iSumCompany:=0;
sClientsName := tblClients['ClientName']+' '+ tblClients['ClientSurname'] ;
sClientID:=tblClients['ClientID'];
tblCollection.First;
while not tblCollection.EOF do
begin
  if (sClientID=tblCollection['ClientID']) AND (strToInt(Copy(tblCollection['CollectionDate'],6,2))=iMonth)
then
```

```
begin
  iSumClient:=iSumClient+(tblCollection['NumberOfCans']);
end;
if strToInt(Copy(tblCollection['CollectionDate'],6,2))=iMonth then
begin
iSumCompany:=iSumCompany+(tblCollection['NumberOfCans']);
end;
tblCollection.Next;
end;
redQ2_2_2.Lines.Add(sClientsName+#13);
redQ2_2_2.Lines.Add('Client collected in month '+intToStr(iMonth)+':'+#9+intToStr(iSumClient));
redQ2_2_2.Lines.Add('Company collected in month
'+intToStr(iMonth)+':'+#9+intToStr(iSumCompany));
redQ2_2_2.Lines.Add('Percentage collected by
client:'+#9+FloatToStrF(iSumClient/iSumCompany*100,ffixed,10,2));

end;

end.
```

```
unit RecycleReceipt_U;

interface

type
  TReceipt = class(TObject)
  private
  var
    fClienID: String;
    fNumOfCans: Integer;
    fKgsOfCardboard: integer;
    fAmount: Real;
  public
    // Provided code
    function getClientID : String;
    function getNumOfCans : Integer;
    function getKgsOfCardboard : integer;
    // Code here
    Constructor create(sClientID:String; iNumCans, iKgsCardboard:integer);
    procedure incNumCans(inum:integer);
    procedure setAmount(rAmount:real);
    function calculateAmount:real;
    function toString:String;
  end;

implementation

uses
  SysUtils, Math;

// =====
// Question 3.1.1
// =====

constructor TReceipt.create(sClientID:String; iNumCans, iKgsCardboard:integer);
begin
  fClienID:=sClientID;
  fNumOfCans:= iNumCans;
  fKgsOfCardboard:=iKgsCardboard;
  fAmount:=0;
end;
```



```
// =====  
// Question 3.1.2  
// =====  
procedure TReceipt.incNumCans(inum: integer);  
begin  
    fNumOfCans:=fNumOfCans+inum;  
end;  
  
// =====  
// Question 3.1.3  
// =====  
Function TReceipt.CalculateAmount:Real;  
var amt1,amt2:real;  
begin  
    amt1:=floor(fKgsOfCardboard/5)*4;  
    case fNumOfCans of  
        1..1500: amt2:=fNumOfCans*0.15;  
        1501..2000: amt2:=1500*0.15+(2000-fNumOfCans)*0.20;  
        2001..2500: amt2:=1500*0.15+500*0.20+(2500-fNumOfCans)*0.22;  
        else amt2:= 1500*0.15+500*0.20+500*0.22+fNumOfCans-2500*0.30;  
    end;  
    result:=amt1+amt2;  
  
    //OR  
    { amt1:=floor(fKgsOfCardboard/5)*4;  
    case fNumOfCans of  
        1..1500: amt2:=fNumOfCans*0.15;  
        1501..2000: amt2:=fNumOfCans*0.20;  
        2001..2500: amt2:=fNumOfCans*0.22;  
        else amt2:= fNumOfCans*0.30;  
    end;  
    result:=amt1+amt2; }  
  
end;  
  
// =====  
// Question 3.1.4  
// =====  
Procedure TReceipt.setAmount(rAmount:real);  
begin  
    fAmount:=rAmount;  
end;
```

```
// =====  
// Question 3.1.5  
// =====  
function TReceipt.toString: String;  
begin  
    setAmount(calculateAmount);  
    result:='Client ID: '+fClientID+#13+'Number of Cans: '+  
    intToStr(fNumOfCans)+#13+'KilogramsOfCardboard: '+  
    intToStr(fKgsOfCardboard)+#13+'Total Amount: '+  
    floatToStrF(fAmount,ffCurrency,10,2);  
end;  
  
// =====  
// Provided code  
// =====  
  
function TReceipt.getClientID: String;  
begin  
    result := fClientID;  
end;  
  
function TReceipt.getNumOfCans: Integer;  
begin  
    result := fNumOfCans;  
end;  
  
function TReceipt.getKgsOfCardboard: integer;  
begin  
    result := fKgsOfCardboard;  
end;  
  
end.
```

```
unit Question3_U;
var
  objReceipt: tReceipt;

// =====
// Question 3.2.1
// =====
procedure TfrmQuestion3.btnQ3_2_1Click(Sender: TObject);
var sID: String;
    iNumCans, iKgs:integer;

begin
  // Provided code
  redQ3.Clear;
  // Question 3.2.1
  sID := cmbQ3_2_1.text;
  iNumCans := sedQ3_2_1_Cans.Value;
  iKgs := sedQ3_2_1_Kgs.value;
  objReceipt := TReceipt.create(sID,iNumCans,iKgs);
  showMessage('Object successfully instantiated');
end;
// =====
// Question 3.2.2
// =====

procedure TfrmQuestion3.btnQ3_2_2Click(Sender: TObject);
begin
  // Provided code
  redQ3.Clear;
  //3.2.2
  redQ3.Lines.Add(objReceipt.toString);
end;
// =====
// Question 3.2.3
// =====

procedure TfrmQuestion3.btnQ3_2_3Click(Sender: TObject);
var inumCans:integer;
begin
  // Provided code
  redQ3.Clear;
```

```

// Question 3.2.3
inumCans:=sedQ3_2_3.Value;
objReceipt.incNumCans(inumCans);
//redQ3.Lines.Add(objPlant.getPlantCode);
redQ3.Lines.Add(objReceipt.toString);

end;
unit Quest4_U;

// =====
// Question 4.1
// =====

procedure TfrmQuestion4.btnQ4_1Click(Sender: TObject);
var r,c: integer;
    sLine: string;
begin
    // Provided code
    redQ4.Clear;
    // Question 4.1
    redQ4.Lines.Add('-----');
    redQ4.Lines.Add('Area'+#9+'Metal'+#9+'Paper'+#9+'Plastic'+#9+'Cardboard');
    redQ4.Lines.Add('-----');

    for r := 1 to 6 do
        begin
            sLine:=arrRegion[r]+#9;
            for c := 1 to 4 do
                begin
                    sLine:=sLine+intToStr(arrCollection[r,c])+#9;
                end;
            redQ4.lines.Add(sLine);
        end;
    end;

// =====
// Question 4.2.1
// =====
procedure TfrmQuestion4.btnQ4_2_1Click(Sender: TObject);
var r,c,temp : integer;
begin
    for r := 1 to 6 do
        begin
            temp:=arrCollection[r,4];
            arrCollection[r,4] := arrCollection[r,3];
            arrCollection[r,3] :=temp;
        end;

```

```

end;

// =====
// Question 4.2.2
// =====

procedure TfrmQuestion4.btnQ4_2_2Click(Sender: TObject);
var Lfile: textfile;
    arrUpdate: array[1..4] of integer;
    j,r,c,p : integer;
    sLine,sArea: string;
begin
    AssignFile(Lfile,'updates.txt');
    reset(Lfile);
    while not eof(Lfile) do
    begin
        c:=0;
        readln(Lfile,sLine);
        sArea:=Copy(sLine,1,1);
        delete(sLine,1,2);
        p:=pos('#',sLine);
        inc(c);
        arrUpdate[c]:=strToint(copy(sLine,1,p-1));
        delete(sLine,1,p);
        p:=pos('#',sLine);
        inc(c);
        arrUpdate[c]:=strToint(copy(sLine,1,p-1));
        delete(sLine,1,p);
        p:=pos('#',sLine);
        inc(c);
        arrUpdate[c]:=strToint(copy(sLine,1,p-1));
        delete(sLine,1,p);
        inc(c);
        arrUpdate[c]:=strToint(sLine);
        for r := 1 to 6 do
        begin
            if pos(sArea,arrRegion[r])>0 then
            begin
                for c := 1 to 4 do
                begin
                    arrCollection[r,c]:=arrCollection[r,c]+arrUpdate[c];
                end;
            end;
        end;
    end;
end;

// Provided code

```

```
procedure TfrmQuestion4.FormShow(Sender: TObject);
begin
  redQ4.Paragraph.TabCount := 4;
  redQ4.Paragraph.Tab[0] := 70;
  redQ4.Paragraph.Tab[1] := 120;
  redQ4.Paragraph.Tab[2] := 170;
  redQ4.Paragraph.Tab[3] := 220;

end;

// =====
// Question 4.3
// =====

procedure TfrmQuestion4.btnQ4_3Click(Sender: TObject);
var iColTot : array[1..4] of integer;
    r,c :integer;
    sLine:string;
begin
  // Question 4.3
  for r := 1 to 4 do
    iColTot[r]:=0;
  for r := 1 to 6 do
  begin
    for c := 1 to 4 do
      begin
        iColTot[c]:=iColTot[c]+arrCollection[r,c];
      end;
    end;
    sLine:='9';
    for r := 1 to 4 do
      begin
        sLine:=sLine+intToStr(iColTot[r])+#9;
      end;
    redQ4.Lines.Add(sLine);

end;

end.
```