education

Department:
Education
REPUBLIC OF SOUTH AFRICA

NATIONAL SENIOR CERTIFICATE

GRADE 12

INFORMATION TECHNOLOGY P1

NOVEMBER 2008

MEMORANDUM

The memorandum consists of 35 pages.

General information:

- **Pages 2 – 12 contain the Delphi Memoranda of possible solutions for Questions 1 to 3 in programming code.**
- **Pages 13 – 25 contain the Java memoranda of possible solutions for questions 1 to 3 in programming code.**
- **Pages 26 – 32 contains Addendums A to G which includes a cover sheet as well as a marking grid for each question for candidates using either one of the two programming languages.**
- **Copies should be made for each learner to be completed during the marking session.**

NOTE: Any workable solution for the questions has to be carefully considered.
There may be alternative solutions to those suggested in the memo provided.

NOTE: Syntax errors must only be penalized when it leads to a logical error in the program up to a maximum of 2 marks per question – see marking grids
Other syntax errors must be ignored.

NOTE: For Question 2 and 3 the half marks must be carried to the coversheet. The final total must be rounded off on the coversheet just below the total.

SECTION A: DELPHI

QUESTION ONE: DATABASE AND PROGRAMMING

DELPHI SOLUTION

```
unit LitterDbaseU;

interface

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

type
  TfrmLitter = class(TForm)
    Panel1: TPanel;
    dbgLitter: TDBGrid;
    Panel2: TPanel;
    btnGauteng: TButton;
    btnDates: TButton;
    btnUpdate: TButton;
    qryLitter: TADOQuery;
    tblLitter: TDataSource;
    btnAllSchools: TButton;
    btnRegMonth: TButton;
    btnPerCapita: TButton;
    btnJuniors: TButton;
    BitBtn1: TBitBtn;
    btnIncomplete: TButton;
    procedure btnGautengClick(Sender: TObject);
    procedure btnDatesClick(Sender: TObject);
    procedure btnUpdateClick(Sender: TObject);
    procedure btnMonthRegisteredClick(Sender: TObject);
    procedure btnIncompleteClick(Sender: TObject);
    procedure btnJuniorsClick(Sender: TObject);
    procedure btnPerCapitaClick(Sender: TObject);
    procedure btnAllSchoolsClick(Sender: TObject);
```

```
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  frmLitter: TfrmLitter;

implementation

{$R *.dfm}

// Question 1.1
procedure TfrmLitter.btnAllSchoolsClick(Sender: TObject);
begin
 qryLitter.Active := false;   ✓          ✓     ✓          ✓
 qryLitter.SQL.Text :='SELECT * FROM SchoolsTb ORDER BY SchoolName ' ;
 qryLitter.Active := true;
end;                                                                    (4)
//=========================================================================
// Question 1.2
procedure TfrmLitter.btnGautengClick(Sender: TObject);
begin
  qryLitter.Active := false;        ✓
  qryLitter.SQL.Text :=  'SELECT  SchoolID, SchoolName,
                                  ✓                        ✓          ✓
           TotalNumLearners FROM SchoolsTb WHERE TotalNumLearners > 500 AND
                                                          ✓     ✓
                                                  SchoolID LIKE "GP%" ' ;
  qryLitter.Active := true;
end;                                                                    (6)

//=========================================================================
// Question 1.3
procedure TfrmLitter.btnIncompleteClick(Sender: TObject);
begin
  qryLitter.Active := False;             ✓                        ✓
  qryLitter.SQL.Text :=  'SELECT SchoolName, Telnum FROM SchoolsTb
                              WHERE TotalNumLearners is NULL' ; ✓
  qryLitter.Active := true;
end;                                                                    (3)
//=========================================================================
// Question 1.4
procedure TfrmLitter.btnJuniorsClick(Sender: TObject);
begin
  qryLitter.Active := false;

                                        ✓              ✓
 qryLitter.SQL.Text :=  'SELECT SUM(LitterWeight)  AS [Litter grade 8 and 9
                                        ✓          ✓     ✓
      learners collected in kg ] FROM LitterTb WHERE (Grade = 8 OR Grade = 9)' ;
 qryLitter.Active := true;
end;                                                                    (5)
//=========================================================================
```

```
// Question 1.5
procedure TfrmLitter.btnMonthRegisteredClick(Sender: TObject);
var
  sName :string;
  numMonth:string;
begin
  qryLitter.Active := False;
  numMonth := InputBox('Enter the number of the month','',''); ✓
                                          ✓
  qryLitter.SQL.Text :=  'SELECT SchoolName, Principal, RegDate, Grade,
                                                  ✓
                 NumLearners FROM SchoolsTb, LitterTb WHERE  ' +
                                 ✓
                        'SchoolsTb.SchoolID = LitterTB.schoolID AND ' +
                                 ✓                  ✓
                        'MONTH(RegDate) = "' + numMonth + '" '  ;
  qryLitter.Active := true;
end;                                                                    (6)
//=========================================================================

// Question 1.6
procedure TfrmLitter.btnDatesClick(Sender: TObject);
begin
  qryLitter.Active := false;
                                 ✓
  qryLitter.SQL.Text := 'SELECT SchoolName, RegDate ' +
                                 ✓                  ✓          ✓
                        'FROM SchoolsTb WHERE RegDate = #2008/05/21# OR Regdate
                                                  = #2008/05/22#' ; ✓
  qryLitter.Active := true;
end;                                                                    (5)
//=========================================================================
// Question 1.7
procedure TfrmLitter.btnUpdateClick(Sender: TObject);
var
  sSchool :string;
  sNumber :string;
begin
  sSchool := InputBox('Which school must be updated?','','') ;   //input ✓
  sNumber := InputBox('Number of learners to be added?','','');
                                 ✓          ✓
  qryLitter.SQL.Text := 'UPDATE SchoolsTb SET TotalNumLearners =
                                 ✓                  ✓
            TotalNumLearners + sNumber WHERE SchoolName = "' +sSchool +'"  ';
  qryLitter.ExecSQL;
  qryLitter.SQL.Text := 'SELECT SchoolName, TotalNumLearners FROM SchoolsTb'; ✓
  qryLitter.Active := true;

end;                                                                    (6)
//=========================================================================
// Question 1.8
procedure TfrmLitter.btnPerCapitaClick(Sender: TObject);
begin
 qryLitter.Active := false;
 qryLitter.SQL.Text :=
                                          ✓                 ✓
            'SELECT SchoolName, Grade,Round(LitterWeight/ NumLearners,2)AS
                                 ✓                  ✓
                       [Litter per Capita] FROM SchoolsTb, LitterTb '
                    + ' WHERE LitterTB.SchoolId = SchoolsTB.SchoolID'; ✓
  qryLitter.Active := true;
end;                                                                    (5)
end.
```

QUESTION 2: OBJECT ORIENTED PROGRAMMING

DELPHI SOLUTION

```
// QUESTION 2.1.1
unit SchoolUxxxx;
interface
uses SysUtils, Dialogs, Math;
  type TSchool = class (TObject) ✓
    private
      fSchoolName  : string; ✓
      fMetalWeight : Real;✓
      fGlassWeight : Real;
      fPaperWeight : Real;    ✓
      fOtherWeight : Real;
    public
      constructor create (sn : string; mw, PW, gw, ow: Real);
      function GetSchoolName : String;
      function toString : string;
     // procedure UpdateWeights (mw, pw, gw, ow: Real);
      function TotalWeight : Real;                    function FundsRaised
(MetalValue, GlassValue, PaperValue, OtherValue:real)
                                                                    :Real;
      function getMetalWeight : Real;
      function getPaperWeight : Real;
      function getGlassWeight : Real;
     end;                                                          (4/2) = 2
//=============================================================================
// QUESTION 2.1.2
implementation

                                      ✓          ✓
constructor TSchool.create(sn: string; mw, pw, gw, ow: Real);
begin
  fSchoolName := sn;    ⎫
  fPaperWeight := pw;   ⎪
  fMetalWeight := mw;   ⎬     ✓✓
  fOtherWeight := ow;   ⎪
  fGlassWeight := gw;   ⎭
end;                                                               (4/2) = 2
//=============================================================================
// QUESTION 2.1.3
function TSchool.toString: string; ✓
begin

 ✓ Result := fSchoolName + #9 + ✓floatToStrF(fMetalWeight,ffFixed,7,2) ✓ +'
Kg'✓ + #9 + FloatToStrf(fPaperWeight, ffFixed, 7, 2) ✓+' Kg'+ #9 +
FloatToStrF(fGlassWeight, ffFixed, 7, 2) ✓ +' Kg' + #9 +
FloatToStrF(fOtherWeight, ffFixed, 7, 2) +'Kg' ✓

end;                                                               (8/2) = 4
//=============================================================================
// QUESTION 2.1.4
function TSchool.TotalWeight: Real; ✓
begin
  Result✓ := fMetalWeight + fPaperWeight + fGlassWeight + fOtherWeight; ✓✓
end;                                                               (4/2) = 2
//=============================================================================
```

```
// QUESTION 2.1.5
function TSchool.FundsRaised (MetalValue, GlassValue, PaperValue, ✓
                                                  OtherValue:real) : Real; ✓
begin
  Result := fMetalWeight * MetalValue +✓
            fPaperWeight  * PaperValue +✓
            fGlassWeight * GlassValue ✓+ FOtherWeight*OtherValue; ✓
end;                                                                    (6/2)=3

//=============================================================================
// QUESTION 2.1.6
function TSchool.getGlassWeight: Real;          ✓ return real types
begin
  Result := fGlassWeight;                       ✓✓all the get methods included
end;
function TSchool.getMetalWeight: Real;          ✓ structure of methods correct
begin
  Result := fMetalWeight;
end;

function TSchool.getPaperWeight: Real;
begin
  Result := fPaperWeight;
end;

function TSchool.GetSchoolName: String; ✓
begin
  Result := fSchoolName; ✓
end;                                                                    (6/2) = 3

end.
//=============================================================================
```

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

type
  TfrmLitterComp = class(TForm)
    MainMenu1: TMainMenu;
    Options1: TMenuItem;
    Displayinformation1: TMenuItem;
    FundsRaised1: TMenuItem;
    Quit1: TMenuItem;
    LookUpaschool1: TMenuItem;
    redOutput: TRichEdit;
    procedure Quit1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Displayinformation1Click(Sender: TObject);
    procedure FundsRaised1Click(Sender: TObject);
    procedure LookUpaschool1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  frmLitterComp: TfrmLitterComp;

implementation
 uses
     SchoolUxxxx; ✓
const
  Metal = 1.50;
  Glass = 120;    ✓✓ // Question 2.2.1
  Paper = 1.00;
  Other = 0.50;
var

  arrSchools   :array[1..50] ✓ of TSchool; ✓                // QUESTION 2.2.2
  noSchools   :integer;
{$R *.dfm}

procedure TfrmLitterComp.Quit1Click(Sender: TObject);
begin
  Application.Terminate;
end;

procedure TfrmLitterComp.FormActivate(Sender: TObject);
var
  tmp, sn, mw, pw, gw, ow: string;
  tf : TextFile; ✓
begin
  AssignFile(tf, 'LitterWeight.txt'); ✓
  if FileExists('LitterWeight.txt') then✓
    begin
      Reset (tf); ✓
    end
    else✓
    begin
      MessageDlg('File "LitterWeight.txt" not found. ✓'+#13+#10+''+#13+#10+
                 'Unable to import data.'+#13+#10+''+#13+#10+
```

```
                'Ensure file "LitterWeight.txt" is in the program folder',
                mtError, [mbOK], 0);
      Exit; ✓
    end;
  noSchools := 0; ✓
  while not Eof (tf) and (noSchools < 50 )do✓
  begin
    Readln (tf, tmp); ✓
    sn := Copy (tmp, 1, Pos ('#', tmp) ✓ - 1) ✓;
    Delete (tmp, 1, Pos ('#', tmp)); ✓
    mw := Copy (tmp, 1, Pos ('#', tmp) - 1);
    Delete (tmp, 1, Pos ('#', tmp));
    pw := Copy (tmp, 1, Pos ('#', tmp) - 1);   ✓✓
    Delete (tmp, 1, Pos ('#', tmp));
    gw := Copy (tmp, 1, Pos ('#', tmp) - 1);
    Delete (tmp, 1, Pos ('#', tmp));
    ow := tmp;

    inc (noSchools); ✓              ✓           ✓
    arrSchools[noSchools] ✓:= TSchool.create(sn,StrToFloat(mw),
            StrToFloat(pw),StrToFloat(gw),StrToFloat(ow)); ✓
  end;

  MessageDlg('Data imported successfully.', mtInformation, [mbOK], 0);
  Closefile (tf); ✓
end;                                                          (24/2) = 12
//============================================================================

procedure TfrmLitterComp.Displayinformation1Click(Sender: TObject);
var
  K         : integer;
  rTotal    :real;
begin                                                  // Question 2.2.3
  redOutput.Paragraph.TabCount := 5;
  redOutput.Paragraph.Tab[1] := 80;
  redOutput.Paragraph.Tab[2] := 100;
  redOutput.Paragraph.Tab[3] := 150;
  redOutput.Paragraph.Tab[4] := 200;
  redOutput.Paragraph.Tab[5] := 270;
  redOutput.Clear;
  redOutput.Lines.Add('Schools and litter weights'); ✓
  redOutput.Lines.Add(' ');

  redOutput.Lines.Add('SchoolName  ' + #9 + 'Metal' + #9 + 'Glass' + #9 +
                                        'Paper' + #9 + 'Other'); ✓
  redOutput.Lines.Add(' ');
  rTotal := 0 ; ✓
  For K := 1 to noSchools do✓
   begin✓
      redOutput.Lines.Add(arrSchools[K].toString); ✓
      rTotal := rTotal + ✓ arrSchools[K].totalWeight; ✓
   end;
   redOutput.Lines.Add(' ');
   redOutput.Lines.Add('The total amount of litter collected is ' + ✓✓
                              FloatToStrF(rTotal, ffFixed,7,2)+ ' kg');

end;                                                          (10/2) = 5
//============================================================================
```

```
procedure TfrmLitterComp.FundsRaised1Click(Sender: TObject);

var
  K         :integer;
  rAmount   :real;
begin
  redOutput.Clear;
  redOutput.Lines.Add('Funds raised during competition'); ✓
  redOutput.Lines.Add(' ');

  For K := 1 to noSchools do✓
   with arrSchools[K] do✓
   begin          ✓
     rAmount:= fundsRaised(Metal, Glass, Paper, Other); ✓✓
     redOutput.Lines.Add(getSchoolName + #9 + FloatToStrF(rAmount,
                                                    ffCurrency,7,2)); ✓✓
    end;
    redOutput.Lines.Add(' ');

end;                                                              (8/2) = 4
//=============================================================================
procedure TfrmLitterComp.LookUpaschool1Click(Sender: TObject);
var
  sName, SchoolName :string;
  K         :integer;
  Found     :boolean;
begin
  sName := InputBox('Enter the name of the school','',''); ✓

  Found := false; ✓
  K := 1; ✓
  redOutput.Lines.Add('Enquiry');
  while not(Found) and (K <= noSchools) do✓
    begin✓
    SchoolName :=  arrSchools[K].getSchoolName; ✓
      if sName =  SchoolName then✓
        begin✓
          redOutput.Lines.Add(SchoolName + ' collected ' + ✓
                        FloatToStr(arrSchools[K].TotalWeight) ✓+ ' kg litter');
          Found := true; ✓
        end
      else
        inc(K); ✓
     end;
   if not(Found) then✓
     redOutput.Lines.Add(sName + ' is not on the list'); ✓
end;                                                              (14/2) = 7

end.
//=============================================================================
```

QUESTION 3: DELPHI - PROGRAMMING

DELPHI SOLUTION:

```
unit Refuse_Uxxxx;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Menus, Grids, StdCtrls, ComCtrls;
type
  TfrmRefuseCollection = class(TForm)
    stgResults: TStringGrid;
    MainMenu1: TMainMenu;
    Options1: TMenuItem;
    Displayinformation1: TMenuItem;
    Calculateaverages1: TMenuItem;
    Displaythewinner1: TMenuItem;
    Quit1: TMenuItem;
    lblHeading: TLabel;
    lblColHeading: TLabel;
    lblAverages: TLabel;
    redOutput: TRichEdit;
    procedure Displayinformation1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Quit1Click(Sender: TObject);
    procedure Calculateaverages1Click(Sender: TObject);
    procedure Displaythewinner1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  frmRefuseCollection: TfrmRefuseCollection;

// QUESTION 3.1.1
implementation
{$R *.dfm}
Var                              ✓ ✓
  arrResultsTable:array[1..7,1..4] of integer;
  arrAverages : array[1..4] of real; ✓
  maxRow:integer; // global ✓
//============================================================================
procedure TfrmRefuseCollection.FormActivate(Sender: TObject); ✓
var
  row, col, no  :integer;
begin
    maxRow := 0; ✓
    for col := 1 to 4 do✓
    begin
      no := 0; ✓
      repeat✓
         no := StrToInt(InputBox('How many school in region? (Max 6) ',
                                          IntToStr(col),'')); ✓✓
       until (no > 0 ) AND (no <= 6); ✓
       if no > maxRow then✓
          maxRow := no; ✓
       for row := 1 to no do✓
        arrResultsTable [row, col] ✓:= Random(700) ✓ + 100; ✓
     end;
end;                                                          (18/2)=9
//============================================================================
```

```
// QUESTION 3.1.2
procedure TfrmRefuseCollection.Quit1Click(Sender: TObject);
begin
  Application.Terminate;
end;

procedure TfrmRefuseCollection.CalculateAverages1Click(Sender: TObject);
var
  row, col, no, total, rowCount :integer;
begin

  for col := 1 to 4 do✓
  begin
   total := 0; ✓
   rowCount := 0; ✓
    for row := 1 to maxRow do✓
     begin
       if arrResultsTable[row, col] <> 0 then✓
         begin✓
           inc(rowCount); ✓
           total := total ✓+ arrResultsTable [row, col]; ✓
         end;
     end;
     arrResultsTable[maxRow+1, ✓ Col] := total; ✓   ✓Correct position

     arrAverages[col] ✓ := total/rowCount; ✓
     end;
end;                                                          (14/2) = 7
//============================================================================
// QUESTION 3.1.3
procedure TfrmRefuseCollection.Displayinformation1Click(Sender: TObject);
var
  Row, Col  :integer;
begin
      for col := 1 to 4 do✓
      arrAverages[col] := 0.0; ✓

  lblHeading.caption := 'Litter collected by schools in 4 regions'; ✓
  lblColHeading.caption := '              Regions'; ✓
  for Col := 1 to 4 do
    stgResults.Cells[Col, 0] := intToStr(Col); ✓
  for row := 1 to maxRow✓ do✓
  begin
    stgResults.Cells[c,row] := 'School ' + IntToStr(row); ✓
    for Col := 1 to 4 do✓
    begin
      stgResults.Cells[Col, Row] ✓ := IntToStr✓ (arrResultsTable[row,Col]); ✓
    end;
  end;
   stgResults.Cells[0, maxRow+1] := 'Totals       '; ✓
  for Col := 1 to 4 do✓
     stgResults.Cells[Col, Row] := IntTostr(arrResultsTable[row,Col]); ✓

  stgResults.Cells[0, maxRow+2] := 'Averages     '; ✓
  for Col := 1 to 4 do✓                                ✓                ✓
     stgResults.Cells[Col, Row+1✓] :=FloatToStrF(arrAverages[Col],ffFixed,10,1);
end;                                                          (20/2) = 10

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

```
// QUESTION 3.1.4
procedure TfrmRefuseCollection.DisplayTheWinner1Click(Sender: TObject);
var
  Row, col, winner, winningSchool, WinningRegion:integer;
begin
   Winner := 0; ✓
   WinningSchool := 0; ✓
   WinningRegion := 0; ✓
   for col := 1 to 4 do✓
   begin
     for row := 1 to maxRow do✓
     begin
       if arrResultsTable[row, col] > Winner then✓
         Begin✓
           winner :=  arrResultsTable[row, col]; ✓
           WinningSchool := row; ✓
           WinningRegion := col; ✓
         end;
      end;
    end;                                              Display outside loop✓
    redOutput.Clear;                                  ✓
    redOutput.Lines.Add('The winner is school ' + IntToStr(WinningSchool) + ' in
                                    region ' + IntToStr(WinningRegion)); ✓
    redOutput.Lines.Add('The winning school collected ' + IntToStr(Winner) + 'kg
                                                          litter'); ✓
end;                                                          (14/2) = 7

// Question 3.2 General marks:
Application.Terminate✓ Naming of components✓                  (4/2) =   2
Clear RichEdit ✓  Code dented in ✓

end.
//============================================================================
```

End of Section A: Delphi

SECTION B: JAVA

QUESTION 1 – JAVA : DATABASE AND PROGRAMMING

Java Question 1 Memorandum: testLitter Class

```
import java.io.*;
import java.sql.*;

public class testLitter
{
  public static void main (String[] args) throws SQLException,IOException
  {
    BufferedReader inKb = new BufferedReader (new InputStreamReader
                                                          (System.in));

    Litter DB = new Litter();
    System.out.println("\f");
    char choice = ' ';
    do
    {
            System.out.println("       MENU");
            System.out.println();
            System.out.println("    A - All Schools");
            System.out.println("    B - Gauteng");
            System.out.println("    C - Incomplete");
            System.out.println("    D - Juniors");
            System.out.println("    E - Month Registered");
            System.out.println("    F - Date Registered");
            System.out.println("    G - Update");
            System.out.println("    H - Litter/Capita");
            System.out.println();
            System.out.println("    Q - QUIT");
            System.out.println(" ");
            System.out.print(" Your Choice? ");
            System.out.println(" ");
            choice = inKb.readLine().toUpperCase().charAt(0);

            switch(choice)
            {
                case 'A':{
                            DB.selectAllSchools();
                            break;
                         }
                case 'B':{
                            DB.selectGauteng();
                            break;
                         }
                case 'C':{
                           DB.selectIncomplete();
                           break;
                          }
                case 'D':{
                           DB.selectJuniors();
                           break;

                          }

                case 'E':{
                           DB.selectMonth();
                           break;
```

```
                    }
                case 'F':{
                            DB.selectDates();
                            break;

                         }
                case 'G':{
                            DB.updateLearners();
                            break;

                         }
                case 'H':{
                            DB.calcLitterPerCapita();
                            break;

                         }

                }
            }
            while (choice != 'Q');

            DB.disconnect();
            System.out.println("Done");
        }
    }
```

Java Question 1 Memorandum: Litter class

```
    import java.sql.*;
    import java.io.*;
     import javax.swing.JOptionPane;

     public class Litter
    {
       Connection conn;


       public Litter ()
       {
       //load the driver
          try
          {
           Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
           System.out.println ("Driver successfully loaded");
          }
          catch (ClassNotFoundException c)
          {
           System.out.println ("Unable to load database driver");
    }
```

Note: Replace the filename-statement with the following statement during testing to avoid having to type in the path every time when executing the program. E refers to the root of the flash drive. Replace the E with the drive where your database has been stored:
String filename = " E:/LitterCompDB.mdb";

```
//connect to the database
         try
         {
            //conn = DriverManager.getConnection ("jdbc:odbc:litter.mdb");
```

```
    System.out.print("Type in the exact location of your database (FOR EXAMPLE –
                                    C:/TEST/LitterCompDB.mdb)");
    BufferedReader inKb = new BufferedReader (new InputStreamReader
                                                          (System.in));

    //String filename = inKb.readLine();
     String filename = "c:/LitterCompDB.mdb";

     String database = "jdbc:odbc:Driver={Microsoft Access Driver
                                                        (*.mdb)};DBQ=";
     database += filename.trim () + ";DriverID=22;READONLY=true}";
     conn = DriverManager.getConnection (database, "", "");

     System.out.println ("Connection to Litter database successfully
                                                          established");

     }
        catch (Exception e)
     {
        System.out.println ("Unable to connect to the database");
       }
      } //end connect
  //=============================================================================
 // QUESTION 1.1
 public void selectAllSchools ()throws SQLException
 {
   System.out.println("\f");
   Statement stmt = conn.createStatement ();
                       ✓          ✓          ✓    ✓
   String sql = "SELECT * FROM SchoolsTb ORDER BY SchoolName";
   ResultSet rs = stmt.executeQuery (sql);
   System.out.printf("%-10s%-27s%-20s%-12s%-20s%-12s","ID","School
     Name","Principal", "TelNumber","Number of learners","Registration Date");
   System.out.println();

  System.out.println("=====================================================");

  while (rs.next ())
   {
    String id = rs.getString ("SchoolID");
    String sName = rs.getString ("SchoolName");
    String sHead = rs.getString ("Principal");
    String sTelnum = rs.getString ("TelNum");
    String Number = rs.getString("TotalNumLearners");
    String sRegDate = rs.getString ("RegDate");
    sRegDate = sRegDate.substring(0,10);

    System.out.printf("%-10s%-27s%-20s%-12s%-20s%-12s",id ,sName,sHead,
                                                  sTelnum,Number,sRegDate);
    System.out.println();
   }
   System.out.println(" ");
   stmt.close ();
 } //select All                                                        (4)
 //=============================================================================
```

```
// QUESTION 1.2
public void selectGauteng ()throws SQLException
 {
  Statement stmt = conn.createStatement ();
                                ✓                                    ✓
   String sql = "SELECT SchoolId, SchoolName, TotalNumLearners FROM SchoolsTb
                                              ✓    ✓          ✓    ✓

                       WHERE TotalNumLearners > 500 AND SchoolId LIKE 'GP%'";
   ResultSet rs = stmt.executeQuery (sql);
   System.out.printf("%-10s%-27s%-20s","ID","School Name","TotalNumLearners");
   System.out.println();

    System.out.println("=====================================================");
   while (rs.next ())
    {
     String id = rs.getString ("SchoolId");
     String sName = rs.getString ("SchoolName");
     String number = rs.getString("TotalNumLearners");
     System.out.printf("%-10s%-27s%-20s",id,sName, number);
     System.out.println();
    }
    System.out.println(" ");
    stmt.close ();
    } //select
                                                                        (6)
   //=========================================================================
   // QUESTION 1.3
   public void selectIncomplete ()throws SQLException
    {
      System.out.println("\f");
      Statement stmt = conn.createStatement ();
                               ✓
      String sql = "SELECT SchoolName, TelNum FROM SchoolsTb WHERE ✓
                                              TotalNumLearners is NULL" ; ✓
      ResultSet rs = stmt.executeQuery (sql);
      System.out.printf("%-20s%-27s","SchoolName","Telnum ");
      System.out.println();
      System.out.println("==============================");
      while (rs.next ())
      {
         String sName = rs.getString ("SchoolName");
         String sTelnum = rs.getString("TelNum");
         System.out.printf("%-20s%-27s",sName,sTelnum);
         System.out.println();
      }
      System.out.println(" ");
      stmt.close ();
      } //                                                               (3)
      //=========================================================================
    // QUESTION 1.4
     public void selectJuniors()throws SQLException
      {
         System.out.println("\f");
         Statement stmt = conn.createStatement ();
                                        ✓              ✓
         String sql = "SELECT SUM(LitterWeight)  AS [Litter grade 8 and 9
                                              ✓              ✓    ✓
            learners collected in kg ] FROM LitterTb WHERE (Grade = 8 OR  Grade
                                                                  = 9) ";
         ResultSet rs = stmt.executeQuery (sql);
         while (rs.next ())
```

```
        {
         String num = rs.getString("Litter grade 8 and 9 learners collected in
                                                                    kg");

         System.out.println("Litter grade 8 and 9 learners collected in kg: " +
                                                                    num);
       }
       System.out.println(" ");
       stmt.close ();
     } //                                                          (5)
     //=========================================================================
// QUESTION 1.5
  public void selectMonth()throws SQLException
    {
      System.out.println("\f");
      Statement stmt = conn.createStatement ();
      String sMonth = JOptionPane.showInputDialog("Enter the number of the
                                                          month ");✓
                                ✓
      String sql = "SELECT SchoolName, Principal, RegDate, Grade, NumLearners
                                ✓                    ✓
           FROM SchoolsTb, LitterTb WHERE SchoolsTb.SchoolID =
                                    ✓                    ✓
           LitterTB.SchoolID  AND MONTH(RegDate) = '"+ sMonth +"' ";

      ResultSet rs = stmt.executeQuery (sql);
      System.out.printf("%-27s%-20s%-20s%-12s%-12s","School
                        Name","Principal","RegDate","Grade","NumLearners");
      System.out.println();

      System.out.println("================================================");
      while (rs.next ())
      {

      String sName = rs.getString ("SchoolName");
      String sHead = rs.getString ("Principal");
      String sDate = rs.getString ("RegDate");
      sDate = sDate.substring(0,10);
      String sGrade = rs.getString ("Grade");
```

The formatting in the printf statement is only available in JDK1.6.0
Download the latest version of idk from the sun website.

```
     String total = rs.getString("NumLearners");

     System.out.printf("%-27s%-20s%-20s%-12s%-12s",sName, sHead,sDate,sGrade,
                                                                    total);
     System.out.println();
     }
     System.out.println(" ");
     stmt.close ();
     } //                                                          (6)
     //=========================================================================
    // QUESTION 1.6
    public void selectDates()throws SQLException
     {
        System.out.println("\f");
        Statement stmt = conn.createStatement ();
                                    ✓                    ✓
        String sql = "SELECT SchoolName,  RegDate FROM SchoolsTb WHERE RegDate
                                ✓       ✓            ✓
                       = #2008/05/21# OR RegDate = #2008/05/22#";
        ResultSet rs = stmt.executeQuery (sql);
```

```
         System.out.printf("%-27s%-27s","School Name","RegDate ");
         System.out.println();
         System.out.println("========================================");
         while (rs.next ())
         {

            String sName = rs.getString ("SchoolName");
            String sDate = rs.getString ("RegDate");
            sDate = sDate.substring(0,10);

            System.out.printf("%-27s%-27s",sName,sDate);
            System.out.println();
         }
         System.out.println(" ");
         stmt.close ();                                                    (5)
      } //=================================================================
      // QUESTION 1.7
      public void updateLearners() throws SQLException
      {
        Statement stmt = conn.createStatement ();
        String sSchool = JOptionPane.showInputDialog("Enter the name of the
                                                      school ");


        String number = JOptionPane.showInputDialog("Enter the number of ✓
                                                     learners to be added ");
        int num = Integer.parseInt(number);
                              ✓                ✓
        String sql = "UPDATE SchoolsTb  SET TotalNumLearners = (TotalNumLearners
        ✓                                  ✓
        + " + num + ") WHERE SchoolName = '"+ sSchool +"' ";

        stmt.executeUpdate (sql); ✓

        sql = "SELECT * FROM SchoolsTb";
        ResultSet rs = stmt.executeQuery (sql);
        System.out.printf("%-10s%-27s%-20s%-12s%-20s%-12s","ID","School
       Name","Principal", "TelNumber","Number of learners","Registration Date");
        System.out.println();

System.out.println("============================================================
==========================================================");

        while (rs.next ())
         {
            String id = rs.getString ("SchoolID");
            String sName = rs.getString ("SchoolName");
            String sHead = rs.getString ("Principal");
            String sTelnum = rs.getString ("TelNum");
            String Number = rs.getString("TotalNumLearners");
            String sRegDate = rs.getString ("RegDate");
            sRegDate = sRegDate.substring(0,10);


            System.out.printf("%-10s%-27s%-20s%-12s%-20s%-12s",id ,sName,sHead,

            sTelnum,Number,sRegDate);
            System.out.println();
         }
         System.out.println(" ");
      stmt.close();
      }                                                                    (6)
      //=================================================================
```

```
// QUESTION 1.8
public void calcLitterPerCapita()throws SQLException
{
  System.out.println("\f");
  Statement stmt = conn.createStatement ();
                                                  ✓
  String sql = "SELECT SchoolName, Grade, Round( LitterWeight / NumLearners
         ✓                ✓                              ✓
         ,2) AS   [Litter per Capita]  FROM SchoolsTb, LitterTb WHERE
               LitterTB.SchoolId = SchoolsTB.SchoolID";✓
  ResultSet rs = stmt.executeQuery (sql);
  System.out.printf("%-27s%-10s%-20s","School Name","Grade","Kg Litter Per
                                                        Capita ");
  System.out.println();
  System.out.println("=====================================================");
  while (rs.next ())
  {
    String sName = rs.getString ("SchoolName");
    String sGrade = rs.getString ("Grade");
    String capita = rs.getString("Litter per Capita");
    System.out.printf("%-27s%-10s%-10s",sName,sGrade,capita);
    System.out.println();
  }
   System.out.println(" ");
   stmt.close ();
} //                                                              (5)
   //=============================================================================

  public void disconnect () throws SQLException
  {
      conn.close ();
  }
  }

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

QUESTION 2 – JAVA: OBJECT-ORIENTED PROGRAMMING

Solution: Java Programming code

```
public class SchoolLitter
 {

   private String fSchoolName;
   private double fMetalWeight = 0;                    // QUESTION 2.1.1
   private double fGlassWeight = 0; ✓✓
   private double fPaperWeight = 0;
   private double fOtherWeight = 0;

                                                       // QUESTION 2.1.2
  public SchoolLitter( String SchoolName, double MetalWeight,double
                 GlassWeight,double PaperWeight,double OtherWeight)
   {
    fSchoolName = SchoolName;
    fMetalWeight = MetalWeight;
    fGlassWeight = GlassWeight;    } ✓✓
    fPaperWeight = PaperWeight;
    fOtherWeight = OtherWeight;
    }                                                          (4/2) = 2
   //=========================================================================
                                                       // QUESTION 2.1.3
   public String✓ toString()
   {                        ✓               ✓               ✓
     String str = fSchoolName + "\t\t" + fMetalWeight + "kg\t" +  fGlassWeight+
            "kg\t"✓ + fPaperWeight + "kg\t"✓ + fOtherWeight+ "kg";✓
     return str;  ✓
   }                                                           (8/2) = 4

  //=========================================================================
                                                       // QUESTION 2.1.4
   public double ✓totalWeight()
    {                          ✓✓
      double total = fMetalWeight + fGlassWeight + fPaperWeight + fOtherWeight;
      return total; ✓
    }                                                          (4/2) = 2
  //=========================================================================
                                                       // QUESTION 2.1.5
  public double ✓fundsRaised(double fMetalValue, double fGlassValue, double
                                        fPaperValue, double fOtherValue) ✓
   {
     double totalFundsRaised =
     this.fPaperWeight * fPaperValue + ✓
     this.fGlassWeight* fGlassValue + ✓
     this.fMetalWeight* fMetalValue +
     this.fOtherWeight * fOtherValue; ✓

     return totalFundsRaised; ✓
   }                                                           (6/2) = 3
   //=========================================================================
                                                       // Question 2.1.6
   public double getfMetalWeight()        ✓✓ all the get methods included
   {
     return fMetalWeight;                 ✓ double return values
   }
   public double getfGlassWeight()        ✓ Structure of the get methods correct
   {
      return  fGlassWeight;
   }
```

```
  public double getfPaperWeight()
   {
      return fPaperWeight;
   }
   public double getfOtherWeight()
   {
      return  fOtherWeight;
   }
   public String ✓getSchoolName()
   {
      return fSchoolName; ✓
   }
  }                                                                    (6/2) = 3
//=============================================================================
```

Java Memorandum: testSchoolLitter class

```
import java.util.Scanner;
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
import java.io.File;
import java.io.FileReader;

public class testSchoolLitter
{
  public static void main (String [] args) throws IOException
{
Scanner input = new Scanner(System.in);

                                                        // QUESTION 2.2.1
double METALVALUE = 1.50;
double GLASSVALUE = 120;     ✓✓
double PAPERVALUE = 1.00;
double OTHERVALUE = 0.50;                                              (2/2) = 1
//=============================================================================
                                                        // QUESTION 2.2.2
SchoolLitter [] arrSchools✓ = new SchoolLitter[200]; ✓
int numSchools = 0; ✓                                   //✓ throws IOException
File inputFile = new File ("LitterWeights.txt");✓
if (inputFile.exists())✓
{
FileReader in = new FileReader (inputFile); ✓
BufferedReader inF = new BufferedReader (in); ✓
String line = inF.readLine ();✓
while (line != null) ✓
{✓
 String[] part = line.split("#");✓✓
 String fName = part[0]; ✓
 double fMetalWeight = Double.parseDouble(part[1]); ✓
 double fGlassWeight = Double.parseDouble(part[2]); ✓
 double fPaperWeight = Double.parseDouble(part[3]);
 double fOtherWeight = Double.parseDouble(part[4]);

 arrSchools [ numSchools] ✓ = new SchoolLitter(fName,fMetalWeight,fGlassWeight,
                              fPaperWeight, fOtherWeight); ✓
 numSchools++;✓

 line = inF.readLine ();✓
 }
 inF.close ();✓                                                        (24/2) = 12
}
else✓
```

```
{
  System.out.println("File does not exist"); ✓
  System.exit(0); ✓
}
System.out.println("\f");
//============================================================================
                                                          // QUESTION 2.2.3
BufferedReader inKb = new BufferedReader (new InputStreamReader (System.in));
 char ch = ' ';
 while (ch != 'Q')
  {
    System.out.println("        Menu ");
    System.out.println("    A - Display information");
    System.out.println("    B - Funds details");
    System.out.println("    C - Total litter enquiry");
    System.out.println("    Q - QUIT");
    System.out.println(" ");
    System.out.print("     Choice? :");
    ch = inKb.readLine().toUpperCase().charAt(0);
    switch (ch)
    {
      case 'A':
      {
       System.out.println("\f");
       double total = 0; ✓
       System.out.println("Schools and Litter Weights");✓
       System.out.println("Name of school      Metal   Glass   Paper   Other");✓
       for (int i = 0; i < numSchools; i++)✓
       {
         System.out.println(arrSchools[i].toString());✓✓
         total = total✓ + arrSchools[i].totalWeight();✓
       }
       System.out.println(" ");                                          ✓
       System.out.println("The total amount of litter collected is "+ total + "
                                                                    kg");✓
       System.out.println("\n\n");
       break;                                                  (10/2) = 5
       }//=========================================================================
      case 'B':
       {
        System.out.println("\f");
        System.out.println("Funds raised during competition");✓
        for (int i = 0; i < numSchools; i++)✓
        {
          double amount =
        ✓                          ✓✓
      arrSchools[i].fundsRaised(METALVALUE,GLASSVALUE,PAPERVALUE,OTHERVALUE);
                                        ✓     ✓               ✓
          System.out.printf("%-25s%2s%8.2f\n",arrSchools[i].getSchoolName(),"R",
                                                          amount);

         }
         System.out.println(" ");
         break;
        }                                                        (8/2) = 4
       //=========================================================================
       case 'C':
       {
          boolean found = false; ✓
          System.out.print("School name? :"); ✓
          String fSchoolName = inKb.readLine();
          System.out.println("\f");
```

```
          System.out.println("Enquiry");
          for (int i = 0; i < numSchools; i++)✓✓
          {
            if (arrSchools[i].getSchoolName()✓.equals(fSchoolName)) ✓
            {✓
                System.out.printf("%-10s%s%6.2f%2s\n", ✓
                  arrSchools[i].getSchoolName()," has collected
                        ",arrSchools[i].totalWeight()✓," kg of litter");✓
                found = true; ✓
                break; ✓
              }
            }
            if (found == false) ✓
              System.out.println(fSchoolName + " was not found");✓
            System.out.println("\n\n");
            break;
           }                                                    (14/2) = 7
           //=====================================================================
         case 'Q':
          {
            System.exit(0);
          }

         }

        }

      }
   }
//=============================================================================
```

QUESTION 3 – JAVA PROGRAMMING

Solution: Java Programming code

```
import java.io.*;
public class testRefuseCollection
 {
  public static void main (String [] args) throws Exception
  {
    RefuseCollection comp = new RefuseCollection();
    comp.populateArray();
    System.out.println("\f");
    BufferedReader inKb = new BufferedReader (new InputStreamReader
    (System.in));
       char ch = ' ';
       while (ch != 'Q')
       {
          System.out.println("        Menu ");
          System.out.println("   A - Display information");
          System.out.println("   B - Calculate averages");
          System.out.println("    C - Display the winner");
          System.out.println("    Q - QUIT");
          System.out.println(" ");
          System.out.print("     Choice? :");
          ch = inKb.readLine().toUpperCase().charAt(0);
                                                          // QUESTION 3.2
        switch (ch)
         {
            case 'A':
               {
               comp.displayArray();✓

                break;
               }
            case 'B':
               {
                 comp.calculateAverages();✓
                 comp.displayArray();✓

                  break;
               }

            case 'C':
               {
                  comp.getWinner();✓
                  break;
               }

            case 'Q':
               {
                  System.exit(0);
               }
             }
       }

    }
  }                                                       (4/2) = 2
//============================================================================
```

Java memorandum: RefuseCollection class

```
import java.io.*;
import java.util.Scanner;
public class RefuseCollection
{                                                                    ✓
                                                                     // QUESTION 3.1.1
   BufferedReader inKb = new BufferedReader (new InputStreamReader (System.in));
      private int[][] arrResultsTable;✓
      private double[] arrAverages; ✓
      int maxRow = 0;

       RefuseCollection()
      {
      }
      public void populateArray()throws Exception✓
      {
          Scanner in = new Scanner(System.in);
          maxRow = 0; ✓
          arrResultsTable = new int[6][4]; ✓
          arrAverages = new double[4]; ✓
          for (int col = 0; col < 4; col++)✓
          {
            int no = 0; ✓
            do✓
            {
              System.out.print("How many schools in region " + (col+1) + "?");✓
               no = Integer.parseInt(inKb.readLine());✓
            }
            while (no > 6); ✓
            if (no > maxRow) ✓
                maxRow = no; ✓
            for (int row = 0; row < no; row++)✓
            {
              arrResultsTable[row][col] ✓ = (int)(Math.random()* 700✓ + 100); ✓
            }
         }
      }                                        (18/2) = 9
    //=========================================================================
                                                                     // QUESTION 3.1.2
       public void calculateAverages()
      {
         int row;
         arrAverages = new double[3];
          for (int col = 0;col < 4; col++)✓
         {
            double countRow = 0; ✓
            int total =0; ✓
            for (row = 0; row < maxRow; row++) ✓✓
            {
                if (arrResultsTable[row][col]!= 0) ✓
               {✓
                  countRow++;✓
                  total = total ✓+ arrResultsTable[row][col]; ✓
                }
            }
            arrResultsTable[row][col] ✓= total ; ✓
            double average = total/countRow; ✓
            arrAverages[col] = average; ✓
           }
           System.out.println("\n\n\n");
      }                                                                (17/2) = 7
   //=========================================================================
```

```
                                                          // QUESTION 3.1.3
    public void displayArray()
     {
      for (int col = 0; col < 4; col++)✓
          arrAverages[col] = 0.0; ✓

          System.out.println("Litter collected by schools in 4 regions");✓
          System.out.println("                        Regions");✓
          System.out.println("              1        2         3         4 ");✓

         for (int row = 0; row < maxRow; row++)✓
         {✓
            System.out.print("School  "+(row + 1)); ✓
             for (int col = 0;col < 4; col++)✓
            {
               System.out.printf("%10d", arrResultsTable[row][col]); ✓
            }
            System.out.println();✓
         }
         System.out.print("Totals   ");✓
         for(int col = 0;col < 4; col++)✓
             System.out.printf("%10d"✓, arrResultsTable[maxRow][col]); ✓
         System.out.println("");✓

         System.out.print("Averages    ");✓
         for(int col = 0;col < 4; col++)✓
             System.out.printf("%9.1f" ✓,arrAverages[col]); ✓
         System.out.println("\n\n");
     }                                                              (20/8) = 10
    //============================================================================

                                                          // QUESTION 3.1.4
      public void getWinner()
     {
        int Winner = 0; ✓
        int WinningSchool = 0; ✓
        int WinningRegion = 0; ✓
        for (int col = 0; col < 4; col++)✓
        {
          for (int row = 0; row < maxRow; row++)✓
           {
              if (arrResultsTable[row][col]> Winner) ✓
              {✓
                 Winner = arrResultsTable[row][col]; ✓
                 WinningSchool = row+1; ✓
                 WinningRegion = col+1; ✓
              }
           }
       }                                                  Display outside loop✓

           System.out.println("The winner is school " + WinningSchool✓ + " in
                                        region " + WinningRegion); ✓
           System.out.println("The winning school collected " + Winner + " kg
                                          litter");✓
           System.out.println("\n\n\n");

     }
   }                                                                 (14/2) = 7
//============================================================================
```

FINAL TOTAL: 120

ADDENDUM A

GRADE 12 NATIONAL SENIOR CERTIFICATE EXAMINATION 2008

INFORMATION TECHNOLOGY PAPER 1

COVER SHEET

Province: _____

Centre Number: _____

Candidate Number: _____

Programming Language (circle the language used): DELPHI/JAVA

Total marks per question		
Question	**Mark out of**	**Learner's mark**
1	**40**	
2	**45**	
3	**35**	
Actual Final Total	**120**	
Final Total rounded off		

ADDENDUM B

QUESTION 1: DELPHI - DATABASE AND PROGRAMMING

Centre Number:……	Candidate Number:……..

QUESTION 1 DELPHI – Marking Grid

Question	Aspects	Max Marks	Learner's Marks
1.1	SELECT * (1) FROM SchoolsTb (1) ORDER BY (1)SchoolName(1)	**4**	
1.2	SELECT SchoolID, SchoolName, TotalNumLearners (1) FROM SchoolsTb (1) WHERE (1) TotalNumLearners > 500(1) AND (1) SchoolID LIKE "GP%" (1)	**6**	
1.3	SELECT SchoolName, Telnum (1) FROM SchoolsTb (1) WHERE TotalNumLearners is NULL (1)	**3**	
1.4	SELECT SUM(LitterWeight) (1) AS [Litter grade 8 and 9 learners collected in kg] (1)FROM LitterTb (1) WHERE (Grade = 8 (1) OR Grade = 9) (1)	**5**	
1.5	SELECT SchoolName, Principal, RegDate, Grade, NumLearners (1) FROM SchoolsTb, LitterTb (1) WHERE SchoolsTb.SchoolID = LitterTB.SchoolID (1) AND (1) MONTH (RegDate) (1) = '" + numMonth + "' (1)	**6**	
1.6	SELECT SchoolName, RegDate (1) FROM SchoolsTb (1) WHERE RegDate = #2008/05/21# (1) OR (1)Regdate = #2008/05/22# (1)	**5**	
1.7	UPDATE SchoolsTb (1) SET (1)TotalNumLearners (1) = TotalNumLearners + sNumber (2) WHERE SchoolName = '" + sSchool +"' (1)	**6**	
1.8	SELECT SchoolName, Grade, Round(LitterWeight/ NumLearners (1), (2) (1) AS [Litter per Capita] (1) FROM SchoolsTb, LitterTb (1) WHERE LitterTB.SchoolId = SchoolsTB.SchoolID (1)	**5**	
	Syntax errors: Penalise only if the syntax error results in a logical error (-1) each, up to maximum of 2 marks		
	SUBTOTAL:	**40**	

ADDENDUM C

QUESTION 2 - DELPHI: OBJECT-ORIENTED PROGRAMMING

Centre Number: _____	**Candidate Number:** _____

QUESTION 2 DELPHI – Marking Grid

Question	Aspects	Max Marks	Learner's Marks
2.1			
2.1.1	**Define TSchool attributes**: Name (1), String (1), All other fields (1) Real (1) **(4/2 = 2)**	**2**	
2.1.2	**Constructor:** Parameters correct order (1), correct types (1) Assignment of fields (2) **(4/2 = 2)**	**2**	
2.1.3	**toString method:** Return String (1), Assign to string to Result/name of function (1), concatenation name(1) and 3 real variables (1) formatted (1) "kg" (1) **(8/2 = 4)**	**4**	
2.1.4	**totalWeight:** Return real (1), Assign value to Result / name of function (1), add all the weights(2) **(4/2 = 2)**	**2**	
2.1.5	**fundsRaised:** Receive four values(1), return real(1), Calculate each weight value(2) Add values(1), Assign to result/name of function. **(6/2=3)**	**3**	
2.1.6	**Accessor methods:** Return type real(1), all the get methods for fields included (2), Structure of methods correct (1), getSchoolName returns a string (1) Assigns SchoolName to result/name of Function **(6/2=3)**	**3**	
2.2			
2.2.1	Assign constant values **(2/2=1)**	**1**	
2.2.2	Uses Object unit (1) Declare array of objects (2) Read from file: Declare text file(1)Assignfile (1) if file exists(1) then Reset (1) else (1) display message(1) and exit (1), Init counter (1) loop (1) Read from file (1), Pos of #(1),Copy (1) without #(1), Delete (1) repeat for other values (1), Inc counter(1) create object (1) with arguments(1) in the correct order (1) and assign to array (1), Close file (1) **(24/2= 12)**	**12**	
2.2.3	**Option on menu:** **Display information**: Heading (1) columns headings (1) init total (1) for loop (2), In loop: call toString method (1) call TotalWeights (1), add weight to total (1) Display message (1) with total formatted (1) **(10/2=5)** **Funds Raised:** Heading(1), loop (1) Inside loop:with object (1) call raisedFunds method (1) with four arguments (1) in correct order(1), display variable(1) formatted (1) **(8/2=4)** **Total litter enquiry**: Ask input(1), Vlag = false (1), counter (1) loop (1 for while or 2 marks for using a for loop), Inside loop(1), Call getSchoolName method (1), Test input against schoolname (1), Inside if (1) Display all the information required (2), Vlag = true / break out of for (1), Else increment counter for (if while was used) (1), Display message if name was not found (2) **(14/2=7)**	**5** **4** **7**	
	Syntax errors: Penalise only if the syntax error results in a logical error (-1) each, up to maximum of 2 marks		
	SUBTOTAL:	**45**	

ADDENDUM D

QUESTION 3 - DELPHI PROGRAMMING

Centre Number: _____	**Candidate Number:** _____		
QUESTION 3 DELPHI : Marking Grid			
Question	**Aspects**	**Max Marks**	**Learners Marks**
3.1.1	In OnActivate / OnCreate EventHandler (1) Declare twodim array (2), declare array/variables for averages (1) globally(1) init maxrow (1) For col (1) init no(1) ask num schools in loop(1) to validate (2) display number of region in the prompt(1)test for maxRow (2) inner for row (1) Random value (2) into array (1) **(18/2)**	**9**	
3.1.2	For Col (1) init counter (1) init total (1) inner for row (1) if to test for a value in the position in array (1) inside if (1) if value add to total (2) and inc counter (1) calculate average outside inner loop (1) assign Total to last row in twodim array (2), assign average to variable/array(1), Assign statements inside outer loop but outside inner loop (1) **(14/2)**	**7**	
3.1.3	Init Twodim array (1), Init averages array/variables(1) Headings (3) For row (1) using MaxRow (1) Display row label(1) inner for col (1) display array elements next to each other (1) blank line for next row/ correct in stringGrid (1) converted to string (1) Display 'Totals' (1) in correct row (1) For loop for totals (2) Display 'Average' (1) in correct row(1) For loop to display averages (1) Formatted with decimal(1) **(20/2)**	**10**	
3.1.4	Init winner (1), init winning region (1) init winning school(1), for col (1), inner for row (1) if to test value against winner (1) inside if (1)assign value to winner(1) assign row to winning school(1), assign col to winning region (1) Display messages(3) outside loops (1) **(14/2)**	**7**	
3.2	Quit: Application.Terminate (1) Descriptive naming of components (1) Clear RichEdit(1) Readable structure – code indented(1) **(4/2)**	**2**	
	Syntax errors: Penalise only if the syntax error results in a logical error (-1) each, up to maximum of 2 marks		
	SUBTOTAL	**35**	

ADDENDUM E

QUESTION 1 – JAVA : DATABASE AND PROGRAMMING

Centre Number: _____	**Candidate Number:** _____

QUESTION 1 JAVA – Marking Grid

Question	Aspects	Max Marks	Learner's Marks
1.1	SELECT * (1)FROM SchoolsTb (1)ORDER BY (1)SchoolName(1)	**4**	
1.2	SELECT SchoolID, SchoolName, TotalNumLearners (1) FROM SchoolsTb (1) WHERE (1) TotalNumLearners > 500(1) AND (1) SchoolID LIKE "GP%" (1)	**6**	
1.3	SELECT SchoolName, Telnum (1) FROM SchoolsTb (1) WHERE TotalNumLearners is NULL(1)	**3**	
1.4	SELECT SUM(LitterWeight) (1) AS [Litter grade 8 and 9 learners collected in kg] (1)FROM LitterTb(1) WHERE (Grade = 8 (1) OR Grade = 9) (1)	**5**	
1.5	Input(1) SELECT SchoolName, Principal, RegDate, Grade, NumLearners (1) FROM SchoolsTb, LitterTb (1) WHERE SchoolsTb.SchoolID = LitterTB.SchoolID (1)AND MONTH(RegDate)(1) = "' + numMonth + "' (1)	**6**	
1.6	SELECT SchoolName, RegDate (1) FROM SchoolsTb (1) WHERE RegDate = #2008/05/21# (1) OR (1)Regdate = #2008/05/22# (1)	**5**	
1.7	Input(1) UPDATE SchoolsTb (1) SET TotalNumLearners(1) = (TotalNumLearners + " + num + ") (1)WHERE SchoolName = "' + sSchool +"' (1) Update(1) Display info	**6**	
1.8	SELECT SchoolName, Grade, Round(LitterWeight / NumLearners (1) , 2) (1) AS [Litter per Capita] (1) FROM SchoolsTb, LitterTb (1) WHERE LitterTB.SchoolId = SchoolsTB.SchoolID(1)	**5**	
	Syntax errors: Penalise only if the syntax error results in a logical error (-1) each, up to maximum of 2 marks		
	SUBTOTAL:	**40**	

QUESTION 2 – JAVA: OBJECT ORIENTED PROGRAMMING ADDENDUM F

Centre Number: -_____	**Candidate Number: _____**

QUESTION 2 JAVA – Marking Grid

Question	Aspects	Max Marks	Learner's Marks
2.1			
2.1.1	Define TSchool attributes: Name (1), String (1), All other fields (1) double (1) **(4/2 = 2)**	**2**	
2.1.2	**Constructor:** Parameters correct order (1), correct types(1) Assignment of fields (2) **(4/2 = 2)**	**2**	
2.1.3	**toString method**: Return String(1), Assign string to return variable (1), concatenation name(1)and 3 double variables (1) formatted(1) "kg" (1) **(8/2 = 4)**	**4**	
2.1.4	**totalWeight**: Return double(1), Assign value to return variable/return(1), add all the weights up(2) **(4/2 = 2)**	**2**	
2.1.5	**fundsRaised:** Receive four values(1), return double(1), Calculate each weight value(2) Add values(1), Assign to return variable/ return **(6/2=3)**	**3**	
2.1.6	**Accessor methods**: Return type double(1), all the get methods for fields included(2), Structure of methods correct(1), getSchoolName returns a string(1) Assigns to return variable/return **(6/2=3)**	**3**	
2.2			
2.2.1	Assign constant values **(2/2=1)**	**1**	
2.2.2	Declare array of objects(2) Read from file: throws IOException(1), Create File object(1) if file exists(1) then create FileReader object(1) and BufferredReader object (1), Init counter(1) while loop (1) Read from file (1), Get pos of #(1),Copy with substring (1) correct values(1), repeat for other values (2) / or 5 marks for any other valid way to extract information from the string, Inc counter(1) create object (1) with arguments(1) in the correct order(1) and assign to array(1), Close file(1), If file does not exist (1)display message(1) and exit(1) **(24/2= 12)**	**12**	
2.2.3	**Option on menu:** **Display information**: Heading(1) columns headings(1) init total(1) loop(2), In loop: call toString method(1) call TotalWeights(1), add weight to total(1), Display message(1) with total formatted(1) **(10/2=5)** **Funds Raised:** Heading(1), loop(1) Inside loop:with object(1) call raisedFunds method(1) with four arguments(1) in correct order(1), display variable(1) formatted(1) **(8/2=4)** **Total litter enquiry**: Ask input(1), Vlag = false(1), counter (1) while(1) / 2 marks for using a for loop, Inside loop(1):Call getSchoolName method (1), Test input against schoolname using equals(1), Inside if(1) Display all the information required(2), Vlag = true(1) / break out of for, Else increment counter for (if while was used)(1)/ break out of for loop(1), Display message if name was not found(2) **(14/2=7)**	**5** **4** **7**	
	Syntax errors: Penalise only if the syntax error results in a logical error (-1) each, up to maximum of 2 marks		
	SUBTOTAL:	**45**	

ADDENDUM G

QUESTION 3: JAVA PROGRAMMING

Centre Number: _____	**Candidate Number: _____**		
QUESTION 3 JAVA - Marking Grid			
Question	**Aspects**	**Max Marks**	**Learners Marks**
3.1			
3.1.1	Throws IOException (1) Create BufferedReader object (1) Declare twodim array (2), declare array/variables for averages (1) init maxrow (1) For col (1) init no (1) ask num schools in loop(1) to validate(2) display number of region as part of prompt(1) test for maxRow (2) inner for row (1) Random value (2) into array (1) **(18/2)**	**9**	
3.1.2	For Col (1) init counter (1) init total(1)inner for row (1) if to test for a value in the position in array (1) inside if (1)if value add to total(2) and inc counter (1) calculate average outside inner loop(1) assign total to last row in twodim array (2), assign average to variable/array (1), Assign statements inside outer loop but outside inner loop(1) **(14/2)**	**7**	
3.1.3	Init averages variables/array (2) Headings (3) For row (1) begin for(1)Display row label(1) inner for col (1) display array elements next to each other(1) blank line for next row (1) Display 'Totals'(1)in correct columns (1) Using Tab or Scanner formatter, For loop for totals(2) Display 'Average' (1) in correct column (1) Using Tab or Scanner formatter, Blank line (1) For loop to display averages (2) **(20/2)**	**10**	
3.1.4	Init winner (1), init winning region (1) init winning school(1), for col(1), inner for row (1) if to test value against winner (1) inside if (1)assign value to winner(1) assign row to winning school (1), assign col to winning region (1) Display messages(3) outside loops(1) **(14/2)**	**7**	
3.2	Display information: Call display method (1) Calculate averages: Call average method (1), call display method (1) Display the winner: Call winner method (1) **(4/2)**	**2**	
	Syntax errors: Penalise only if the syntax error results in a logical error (-1) each, up to maximum of 2 marks		
	SUBTOTAL	**35**	