



Province of the
EASTERN CAPE
EDUCATION



Downloaded from: www.mycourses.co.za

NATIONAL SENIOR CERTIFICATE

GRADE 12

SEPTEMBER 2022

INFORMATION TECHNOLOGY P1 MARKING GUIDELINE

MARKS: 150

This marking guideline consists of 16 pages.

NAME OF LEARNER:				
TOTAL QUESTION 1:	TOTAL QUESTION 2:	TOTAL QUESTION 3:	TOTAL QUESTION 4:	TOTAL
/40	/40	/40	/30	/150

QUESTION 1: GENERAL PROGRAMMING SKILLS		MAX. MARKS	MARKS ACHIEVED
1.1	Button [1.1 Shape] shpTest.Shape := stcircle; ✓ shpTest.Brush.Color := clgreen; ✓ gbxQuestion12.Visible := true; ✓ gbxQuestion13.Enabled := true; ✓	4	
1.2	Button [1.2 Decode] Get input from edtInput ✓ Set new string to empty string ✓ Loop until input string is empty ✓ Copy ordinal value and convert to integer ✓ Delete ordinal value from input string ✓ Convert ordinal value to character using CHR ✓ Add character to new string ✓ Display new string in edtOutput ✓	8	
1.3	Button [1.3 Test] Clear the list box ✓ Get input from spinedit ✓ Initialise array counter ✓ <i>Get the number of prime factors from the input and store in array – 7 marks:</i> Outer Loop from 1 to the input ✓ Set number of prime factors to 0 ✓ Inner Loop from 1 to outer loop counter (nested loop) ✓ If inner loop counter is a prime factor ✓ Then increase number of prime factors by one ✓ If number of prime factors = 2 ✓ Then add outer loop counter to an array ✓		

QUESTION 2: DATABASE PROGRAMMING		MAX. MARKS	MARKS ACHIEVED
2.1.1	Button: [Q 2.1.1] 'select DeliveryAddress, DeliveryDate from tblorders order by deliverydate Desc'	3	
	Concepts: SELECT two correct fields ✓ FROM correct table ✓ ORDER BY correct field DESC ✓		
2.1.2	Button: [Q 2.1.2] 'select count(orderid) as Orders20orMore from tblorders where items >= 20'	4	
	Concepts: SELECT count(any field from Orders table) ✓ as Orders20orMore ✓ FROM correct table ✓ WHERE items >=20 ✓		
2.1.3	Button: [Q 2.1.3] 'select * from Orders where Deliveryaddress like ' + quotedstr('%' + sline + '%')	4	
	Concepts: SELECT * (all fields) ✓ FROM correct table ✓ WHERE DeliveryAddress LIKE ✓ quotedstr('%' + sline + '%') ✓		
2.1.4	Button: [Q 2.1.4] 'update orders set collect = true where deliverydate is null'	3	
	Concepts: Update tblOrders ✓ Set collect = true ✓ Where DeliveryDate is null ✓		
2.1.5	Button: [Q 2.1.5] 'select DeliveryAddress, deliverydate - orderdate as DaysToDeliver from tblOrders where collect = false'	4	
	Concepts: Select DeliveryAdress ✓ DeliveryDate – OrderDate ✓ From Orders ✓ Where collect = false ✓		

2.1.6	Button: [Q 2.1.6] 'select ShopName, format(sum(orderamount,"currency") as TOTALSales from tblorders, tblshop where tblshop.shopid = tblorders.shopid group by shopname		
	Concepts: Select ShopName, ✓ format(sum(OrderAmount ✓, "currency") ✓ as TOTALSales ✓ from tblOrders, tblShop ✓ where tblShop.ShopID = tblOrders.ShopID ✓ group by ShopName ✓	7	
	2.1 Subtotal: SQL	25	
2.2.1	Button: [Q 2.2.1] tblShop.First ✓ while not tblShop.eof do ✓ if tblShop['Online'] = true then ✓ redOutput.Lines.Add(tblShop['ShopName']) ✓ tblShop.Next ✓	5	
2.2.2	Button: [Q 2.2.2] Go to first row of tblshop table and loop ✓ if ShopName = combobox item ✓ then set integer variable to ShopID ✓ go to next row of tblshop table set real sum variable = 0 ✓ Go to first row of tblorders table and loop If ShopID = integer variable from first loop ✓ Then display in redoutput: OrderDate converted to string ✓ and OrderAmount in currency format ✓ Add OrderAmount to real sum variable ✓ Go to next row of tblOrders table ✓ After loop display real sum variable in richedit in currency format ✓	10	
	2.2 Subtotal: Code constructs	15	
	TOTAL QUESTION 2	40	

QUESTION 3: OBJECT-ORIENTATED PROGRAMMING		MAX. MARKS	MARKS ACHIEVED
3.1.1	Constructor Create: Correct name ✓ with one string parameter and one integer parameter ✓ Set attributes to correct parameter values ✓ Set fCost = 70 * fHours ✓	4	
3.1.2	procedure setTotalCost Correct method – procedure ✓ One integer parameter ✓ Set correct attribute (fTotalCost) ✓ = 70 * parameter ✓	4	
3.1.3	Function CalculateDays(hours : integer) : integer Correct method – integer function ✓ one integer parameter ✓ If hours > 8 ✓ then calculate and return correct days (use Ceil or other method) ✓ else ✓ return 1 day ✓	6	
3.1.4	function toString: string; correct string method ✓ correct attributes returned as result ✓ use #13, ✓ converted hours and cost (currency) to string ✓	4	
2.1 Subtotal: Object class		18	
3.2.1	Button [Q 3.2.1 Add to quote] Get inputs from spinedit ✓ and list box ✓ Instantiate the object Object name = ✓ tRepair.create ✓ one string, one integer parameter ✓ In correct order ✓ Use method of class ✓ and object name to display ✓ Display empty line in redQ3 ✓ Use method of class and object name ✓ to add up total hours global variable ✓	11	
3.2.2	Button [Q 3.2.2 Finalise Quote] Use object name and method of the class ✓ to settotalcost with global total hours as parameter ✓ Use object name and method of the class ✓ to gettotalcost and set local variable ✓ If cbxparts is checked ✓ then add 150 to total cost ✓ Display days in richedit using object name ✓ and correct method calculatedays with itotalhours as parameter ✓ converted to string ✓ Display total cost in richedit ✓ converted to currency ✓	11	
2.2 Subtotal: Form class		22	
TOTAL QUESTION 3		40	

QUESTION 4: PROBLEM-SOLVING		MAX. MARKS	MARKS ACHIEVED
4.1	Assignfile statement ✓ Reset text file for reading ✓ Set column counter to 0 ✓ Loop through the text file ✓ Read a line ✓ Increment column ✓ Set row = 1 ✓ Add line to array, ✓ converted to integer ✓ Read a line ✓ Set row = 2 ✓ Add line to array, converted to integer ✓	12	
4.2	Set counters for A and B to 0 ✓ Set column counter to 0 ✓ Loop through array ✓ Set row = 1 ✓ Add 1 to column counter ✓ Set first number to array contents using column and row counters ✓ [row, column] Set row = 2 ✓ Set second number to array contents using column and row counters ✓ [row, column] if first number > second number ✓ then display in richedit correct wording – district number and A ✓ increment A counter ✓ else ✓ display in richedit correct wording – district number and B ✓ increment B counter ✓ if A counter > B counter ✓ then display winner A ✓ else ✓ display winner B ✓	18	
TOTAL QUESTION 4		30	

SAMPLE SOLUTIONS**QUESTION 1**

```
procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);
var inum, i1,i2, k, x, m, ifactors : integer;
arrprime : array[1..100] of integer;
icount: integer;
bprime : boolean;
begin
Istoutput.clear;

icount := 0;
inum := sednumber.Value;

for k := 1 to inum do
begin
ifactors := 0;
for m := 1 to k do
begin
if k mod m = 0 then
inc(ifactors);
end;
if ifactors = 2 then
begin
inc(icount);
arrprime[icount] := k;
end;
end;

bprime := false;
for k := 1 to icount - 1 do
begin
i1 := arrprime[k];
for x := k + 1 to icount do
begin
i2 := arrprime[x];
if i1 + i2 = inum then
begin
Istoutput.Items.Add(inttostr(i1) + ' + ' + inttostr(i2));
bprime := true;
end;
end;
end;

if bprime = false then
showmessage(inttostr(inum) + ' cannot be calculated using the sum of two prime
numbers')
end;
```



```
procedure TfrmQuestion1.btnQ1_4Click(Sender: TObject);
var rsq, rc : real;
rr : real;
begin
rsq := 0.2 * 0.2;
rc := Pi * (0.9/(2 * Pi)) * (0.9/(2 * Pi));
rr := (rc - rsq) * 100;
lbloutput.Caption := lbloutput.Caption + floattostfrf(rr,ffixed,8,3) + ' square cm';
end;
```

```
procedure TfrmQuestion1.btnQ1_1Click(Sender: TObject);
begin
shptest.Shape := stcircle;
shptest.Brush.Color := clgreen;
gbquestion1_2.Visible := true;
gbquestion1_3.Enabled := true;
end;
```

```
procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject);
var sline : string;
snew , sword : string;
k, inum : integer;
begin
// If text in edtinput is erased use the following as input:
//      '73 84 32 105 115 32 101 97 115 121 33'
```

```
sline := edtinput.Text;
snew := "";
while length(sline) <> 0 do
begin
sword := copy(sline,1, pos(' ',sline) - 1);
delete(sline,1, pos(' ',sline));
inum := strtoint(sword);
snew := snew + chr(inum);
end;
edtoutput.text := snew;
end;

end.
```

QUESTION 2

```
procedure TQuestion_2.btnQuestion2_1_1Click(Sender: TObject);
// Provided code - do not change
var
  sSQL1: String;
begin
/// enter your code below//
  sSQL1 := 'select DeliveryAddress, DeliveryDate from orders order by deliverydate desc';

  // Provided code - do not change
  dbCONN.runSQL(sSQL1);
  if length(ssql1) <> 0 then
    dbconn.setgridforsql1(dbgsq1);

end;

procedure TQuestion_2.btnQuestion2_1_2Click(Sender: TObject);
// Provided code - do not change
var
  sSQL2: String;
begin
/// enter your code below//
  sSQL2 := 'select count(orderid) as Orders20orMore from orders where items >= 20';

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

end;

procedure TQuestion_2.btnQuestion2_1_3Click(Sender: TObject);
// Provided code - do not change
var
  sline : string;
  sSQL3: String;
begin
  sline := inputbox('Enter a street name','Nxolo Street');
/// enter your code below//
  sSQL3 := 'select * from Orders where Deliveryaddress like ' + quotedstr('%' + sline +
'%');

  // Provided code - do not change
  dbCONN.runSQL(sSQL3);
  if length(ssql3) <> 0 then
    dbconn.setgridforsql3(dbgsq1);
end;

procedure TQuestion_2.btnQuestion2_1_4Click(Sender: TObject);
// Provided code - do not change
var
  sSQL4: String;
begin
/// enter your code below//
  sSQL4 := 'update orders set collect = true where deliverydate is null';
```

```
// Provided code - do not change
dbCONN.executeSQL(sSQL4,dbgshop,dbgorder,dbgsql);
  if length(ssql4) <> 0 then
    dbconn.setgridforsql3(dbgsql);
end;
```

```
procedure TQuestion_2.btnQuestion2_1_5Click(Sender: TObject);
// Provided code - do not change
var
  sSQL5: String;
begin
  /// enter your code below//
  sSQL5 := 'select DeliveryAddress, deliverydate - orderdate as DaysToDeliver from
Orders where collect = false';
```

```
// Provided code - do not change
dbCONN.runSQL(sSQL5);
if length(ssql5) <> 0 then
  dbconn.setgridforsql5(dbgsql);
end;
```

```
procedure TQuestion_2.btnQuestion2_1_6Click(Sender: TObject);
// Provided code - do not change
var
  sSQL6: String;
begin
  /// enter your code below//
  sSQL6 := 'select ShopName, format(sum(orderamount),"currency") as TOTALSales from
orders, shop where shop.shopid = orders.shopid group by shopname';
```

```
// Provided code - do not change
dbCONN.runSQL(sSQL6);
  if length(ssql6) <> 0 then
    dbconn.setgridforsql6(dbgsql);
end;
```

```
procedure TQuestion_2.btnQuestion2_2_1Click(Sender: TObject);
begin
// Provided code - do not change
redoutput.Clear;
redoutput.Lines.Add('Shops that are available for online shopping');
redoutput.Lines.Add('-----');
/// enter your code below//
tblshop.First;
while not tblshop.eof do
  begin
    if tblshop['Online'] = true then
      redoutput.Lines.Add(tblshop['Shopname'] ) ;
    tblshop.Next;
  end;
end;
```

```
procedure TQuestion_2.btnQuestion2_2_2Click(Sender: TObject);
var sshop : string; // Provided code - do not change
bfound : boolean;
inum : integer;
rsum : real;
begin
// Provided code - do not change
redoutput.Clear;
redoutput.Lines.Add('Order Date' + #9 + 'Amount of order');
redoutput.Lines.Add('-----');
sshop := cmbshop.Text;
/// enter your code below//
bfound := false;
tblshop.First;
while (not tblshop.Eof) and (bfound = false) do
begin
if tblshop['shopname'] = sshop then
begin
inum := tblshop['shopid'];
bfound := true;
end;
tblshop.Next;
end;
rsum := 0;
tblorders.first;
while not tblorders.eof do
begin
if tblorders['shopid'] = inum then
begin
redoutput.Lines.Add(datetostr(tblorders['Orderdate']) + #9 +
floattostf(tblorders['orderamount'],ffcurrency,8,2));
rsum := rsum + tblorders['orderamount'];
end;
tblorders.Next;
end;
redoutput.Lines.Add("");
redoutput.Lines.Add('Total amount ordered: ' + floattostf(rsum,ffcurrency,8,2));

end;
```

QUESTION 3**Class Unit:**

unit Question3ClassDefinition;

interface

uses sysutils, math, dialogs;

type

TRepair = class

private

frepairname : string;

fhours : integer;

fcost : real;

ftotalcost : real;

public

constructor create(sname : string; ihrs : integer);

function calculatedays(ihrs : integer) : integer;

function gethours : integer;

function tostring : string;

procedure settotalcost(ihrs : integer);

function gettotalcost : real;

end;

implementation

{ TRepair }

constructor TRepair.create(sname: string; ihrs: integer);

begin

frepairname := sname;

fhours := ihrs;

fcost := 70 * fhours;

end;

// provided code do not delete//

function TRepair.gethours: integer;

begin

result := fhours;

end;

// provided code do not delete//

function TRepair.gettotalcost: real;

begin

result := ftotalcost;

end;

procedure TRepair.settotalcost(ihrs: integer);

begin

ftotalcost := ihrs * 70;

end;

```
function TRepair.calculatedays(ihrs: integer) : integer;
var idays : integer;
begin
  if ihrs > 8 then
    begin
      idays := ceil(ihrs/8);
    end
  else
    idays := 1;
  result := idays;
end;
```

```
function TRepair.tostring: string;
begin
  result := frepairname +
    #13 + 'Hours to complete: ' + inttostr(fhours) +
    #13 + 'Labour Cost: ' + floattostrf(fcost,ffcurrency,8,2)
end;
```

end.

Main Unit:

```
procedure TForm1.btnQ3_2_1Click(Sender: TObject);
begin
  objrepair := trepair.create(lstrepairs.Items[lstrepairs.ItemIndex],sedhours.value);
  redq3.Lines.Add(objrepair.tostring);
  redq3.Lines.Add("");
  itotalhours := itotalhours + objrepair.gethours;
end;
```

```
procedure TForm1.cbxBuyPartsClick(Sender: TObject);
var rcost : real;
begin
  objrepair.settotalcost(itotalhours);
  rcost := objrepair.gettotalcost;
```

```
if cbxparts.checked = true then
  rcost := rcost + 150;
```

```
redq3.Lines.Add("");
redq3.Lines.Add('Total days to complete all jobs = ' +
  inttostr(objrepair.calculatedays(itotalhours)));
redq3.Lines.Add('Total labour cost = ' + floattostrf(rcost,ffcurrency,8,2));
end;
```

QUESTION 4

```
unit Question4_u;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Grids, ComCtrls;

type
  TForm1 = class(TForm)
    pgc: TPageControl;
    tbsvoting: TTabSheet;
    btnQ4_1: TButton;
    btnQ4_2: TButton;
    redQ4: TRichEdit;
    procedure btnQ4_1Click(Sender: TObject);
    procedure btnQ4_2Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  arrvotes : array[1..2,1..10] of integer;

implementation
{$R *.dfm}

procedure TForm1.btnQ4_1Click(Sender: TObject);
var myfile : textfile;
    irow, icol, inum: integer;
    sline : string;
begin
  assignfile(myfile, 'votes.txt');
  reset(myfile);
  icol := 0;
  while not eof(myfile) do
    begin
      readln(myfile,sline);
      inc(icol);
      irow := 1;
      arrvotes[irow,icol] := strtoint(sline);
      readln(myfile,sline);
      irow := 2;
      arrvotes[irow, icol] := strtoint(sline);
    end;
end;
```

```
procedure TForm1.btnQ4_2Click(Sender: TObject);
var
irow, icol, inum1, inum2, k, ia, ib : integer;
begin
  icol := 0;
  ia := 0;
  ib := 0;
  while icol < 10 do
  begin
    irow := 1 ;
    inc(icol);
    inum1 := arrvotes[irow,icol] ;
    irow := 2;
    inum2 := arrvotes[irow,icol];
    if inum1 > inum2 then
      begin
        redq4.Lines.Add('District '+ inttostr(icol) + ' winner is A');
        inc(ia);
      end
    else
      begin
        redq4.Lines.Add('District '+ inttostr(icol) + ' winner is B');
        inc(ib);
      end;
    end;
    redq4.Lines.Add("");
  if ia > ib then
    redq4.Lines.Add('The overall winner is A')
  else
    redq4.Lines.Add('The overall winner is B')
  end;

end.

end.
```