higher education & training

Department:
Higher Education and Training
REPUBLIC OF SOUTH AFRICA

MARKING GUIDELINE

NATIONAL CERTIFICATE (VOCATIONAL)

PRINCIPLES OF COMPUTER PROGRAMMING
(Second Paper)
NQF LEVEL 3

23 NOVEMBER 2018

This marking guideline consists of 13 pages.

Copyright reserved

QUESTION 1

1.1
```
Option Explicit On
Option Strict On

Public Class frmHomeLoan

 'Global Variables
   Dim dblPrincipal As Double = 0D
   Dim dblAnnualInterestRate As Double = 0D
   Dim intNumOfMonths As Integer = 0I
```

Variable Declaration {All THREE variables}✓
Variables Initialised✓ (2)

1.2 1.2.1
```
'QUESTION 1.2.1
Private Sub GetInput(ByRef dblPrincipal As Double, ByRef
dblAnnualInterestRate As Double, ByRef intNumOfMonths As
Integer)
```

Restricted Access Modifier {"Private"}✓
Type of procedure {"Sub"}✓
Three parameters/arguments ✓
Correct parameter passing {"ByRef"}✓ (4)

1.2.2
```
'QUESTION 1.2.2
 dblPrincipal = CDbl(txtPrincipalAmount.Text) -
CDbl(txtDeposit.Text)
 dblAnnualInterestRate = CDbl(txtInterestRate.Text)/100
 intNumOfMonths = CInt(tkbRepayment.Value) * 12
End Sub
```

All three assignments {=}✓
Calculation of Principal amount {"txtPrincipal.Text – txtDeposit.Text"}✓
Calculation of Annual Interest Rate✓
Calculation of Number of Months✓
Conversion Function {"CDbl/CInt/Val"}✓ (5)

1.3 1.3.1
```
'QUESTION 1.3.1
Private Sub Repayment(ByVal dblPrincipal As Double, ByVal
dblAnnualInterestRate As Double, ByVal intNumOfMonths As
Integer)
```

Restricted Access Modifier {"Private"}✓
Type of procedure {"Sub"} ✓
Three parameters/arguments ✓
Correct parameter passing {"ByVal"}✓ (4)

1.3.2
```
'QUESTION 1.3.2
 Dim dblMonthlyInterestRate As Double
 Dim dblMonthlyInstalment As Double
 Dim dblTotalPayable As Double
 Dim dblTotalInterestPayable As Double
```

Declared FOUR variables✓
Appropriate datatype {"Double"}✓ (2)

1.3.3

```
'QUESTION 1.3.3
 dblMonthlyInterestRate = dblAnnualInterestRate / 12

 dblMonthlyInstalment = -Pmt(dblMonthlyInterestRate,
intNumOfMonths, dblPrincipal)

 dblTotalPayable = dblMonthlyInstalment * intNumOfMonths

 dblTotalInterestPayable = dblTotalPayable - dblPrincipal
```

Calculate Monthly Interest Rate ✓✓
Calculate Monthly Instalment ✓✓
Calculate Total Payable ✓✓
Calculate Total Interest Payable ✓✓ (8)

1.3.4

```
'QUESTION 1.3.4
 lstDisplay.Items.Clear()
 lstDisplay.Items.Add("Your monthly repayments will be:")
 lstDisplay.Items.Add(dblMonthlyInstalment.ToString("C2") &
vbCrLf)
 lstDisplay.Items.Add("")

 lstDisplay.Items.Add(vbCrLf & "The total amount payable
 after " & intNumOfMonths & " months is :")

 lstDisplay.Items.Add(dblTotalPayable.ToString("C2") &
 vbCrLf & vbCrLf)

 lstDisplay.Items.Add("")

 lstDisplay.Items.Add(vbCrLf & "The total amount of
 interest payable after " & intNumOfMonths & " months is
 :")


 lstDisplay.Items.Add(dblTotalInterestPayable.ToString("C2"
 ) & vbCrLf)

End Sub
```

Cleared list box control ✓
Display output ✓
Formatting string output {.ToString("C2")} ✓ (3)

1.4

```
'QUESTION 1.4
Private Sub tkbRepayment_Scroll(…) Handles tkbRepayment.Scroll

   lblRepaymentPeriod.Text = tkbRepayment.Value & " years" ✓

End Sub
```

(1)

1.5 1.5.1 (a)

```
'QUESTION 1.5
Private Sub btnCalcRepayment_Click(…) Handles
btnCalcRepayment.Click

'QUESTION 1.5.1
'(a)
  If IsNumeric(txtPrincipalAmount.Text) = False Then
    MessageBox.Show("The principal amount cannot be
     alphanumeric", "Input Data Validation")
    txtPrincipalAmount.Focus()
    Exit Sub
  End If
```

(b)

```
  If CDbl(txtInterestRate.Text) < 10.25 Then
    MessageBox.Show("Interest Rate cannot be less than
     the Prime Lending Rate", "Input Data Validation")
    txtInterestRate.Focus()
    Exit Sub
  End If
```

(c)

```
  If tkbRepayment.Value = 0 Then
    MessageBox.Show("Use the trackbar to select a
     repayment period.", "Input Data Validation")
    tkbRepayment.Focus()
    Exit Sub
  End If
```

All THREE validation criteria✓
Relevant control focused {All THREE}✓ Terminate Statement ("Exit Sub") {All THREE}

(3 × 1) (3)

1.5.2

```
'QUESTION 1.5.2
 If txtDeposit.Text = "" Then
    txtDeposit.Text = "0"
 End If
```

Correct value {0} assigned to text box✓ (1)

1.5.3

```
'QUESTION 1.5.3
   Call GetInput(dblPrincipal, dblAnnualInterestRate,
intNumOfMonths)
   Call Repayment(dblPrincipal, dblAnnualInterestRate,
                                                intNumOfMont
                                                hs)
End Sub
```

Invoke Procedures {Both}✓✓

NOTE: The *Call* keyword is not necessary.
Correct Parameters/Arguments passed {Both}✓✓ (4)

1.6.1

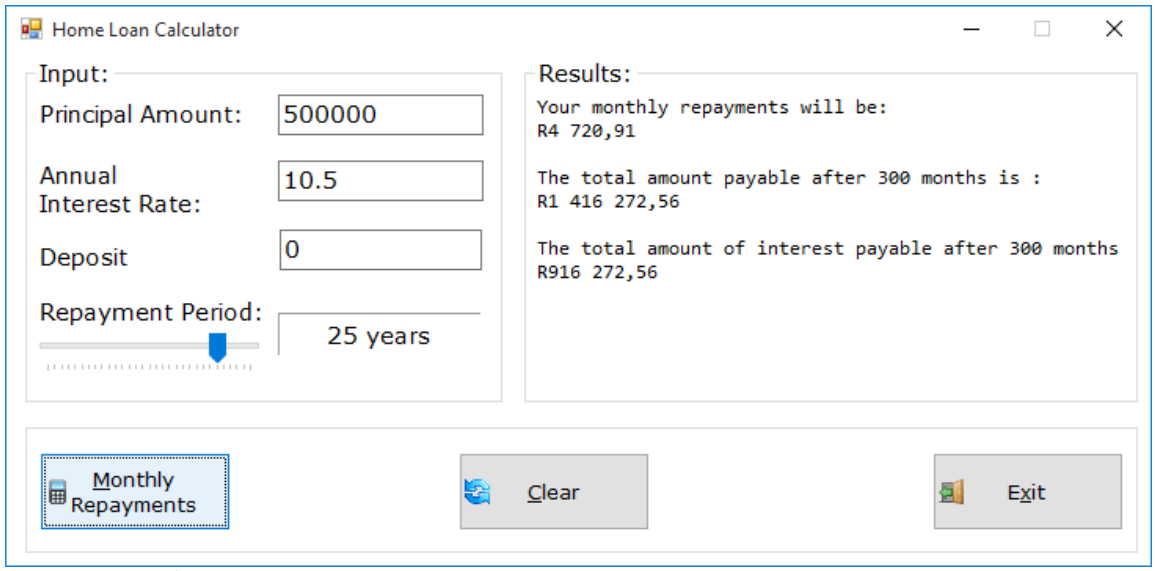
```
'QUESTION 1.6
Private Sub btnClear_Click(…) Handles btnClear.Click

        lstDisplay.Items.Clear()
        txtPrincipalAmount.Clear()
        txtDeposit.Clear()
        txtInterestRate.Clear()
```

Correct method {.Clear( )}✓

1.6.2
```
        tkbRepayment.Value = 0
        lblRepaymentPeriod.Text = "0 Years"
```
Label control AND track bar control set to default values✓
```
End Sub
```
(2 × 1) (2)

1.7 Printout of source code

1.8



Correct output✓ (1)

[40]

QUESTION 2

2.1
```
Option Explicit On
Option Strict On

Public Class frmLanguage
'QUESTION 2.1 Declare one-dimensional array containing subject names

    Dim strArrSubjectNames() As String = {"ENGLISH FAL P2", "SEPEDI
                                          FAL P2"}
```
Array declared and initialised correctly✓ (1)

2.2
```
'QUESTION 2.2
    Dim intCounterVariable As Integer
    Dim intNoSectionsAnswered As Integer = 0
    Dim intNumberOfCandidates As Integer = 0
```
Variable Declaration {All THREE variables} (1)

2.3
```
'QUESTION 2.3 Form load subprocedure
 Private Sub frmLanguage_Load(…) Handles MyBase.Load
    Dim index As Integer = 0
'Q2.3.1 Populate the combo box
    Do Until index > (strArrSubjectNames.Length - 1)
        cboSubjectNameList.Items.Add(strArrSubjectNames(index))
        index = index + 1
    Loop
```
Correct loop condition✓✓
Array element added to combo box✓
Increment loop counter ✓ (4)

2.4
```
'QUESTION 2.4
 Private Sub cboSubjectNameList_SelectedIndexChanged(…) Handles
      cboSubjectNameList.SelectedIndexChanged

  btnEnter.Enabled = True
  Call DisplaySelectedSubject()
```

2.4.1
```
'QUESTION 2.4.1
  intNumberOfCandidates = CInt(InputBox("Enter the number
                               of candidates for the exam",
                               "Input Entry"))
```
Input dialog box ✓
Variable assignment ✓

2.4.2
```
'QUESTION 2.4.2
   'Redimension all the arrays
    ReDim strArrExamNumber(intNumberOfCandidates - 1)
    ReDim intArrSecAMark(intNumberOfCandidates - 1)
    ReDim intArrSecBMark(intNumberOfCandidates - 1)
    ReDim intArrSecCMark(intNumberOfCandidates - 1)
    ReDim intArrSecDMark(intNumberOfCandidates - 1)
    ReDim intArrTotalMarks(intNumberOfCandidates - 1)
    ReDim strArrMessage(intNumberOfCandidates - 1)
```
Dimensioning all arrays {ReDim}✓
Adjust Index {intNumberOfCandidates - 1} ✓

```
  'Enable the input group box
    grpInputData.Enabled = True

  'Focus the examination number text box
    txtExamNumber.Focus()
```

(2 × 2) (4)

2.5
```
'QUESTION 2.5

Private Sub ValidateExamNo()
  If txtExamNumber.Text = "" Then
```
Conditional Statement {"IF/End IF"} ✓
```
     MessageBox.Show("Surname field cannot be left blank.", "Input
                     Data Validation")
     txtExamNumber.Focus()
  End If
End Sub
```
MessageBox ✓
Focus control ✓ (3)

2.6
```
'QUESTION 2.6
Private Function ValidateMark(ByVal value As String) As Boolean
   Return Type of Function {"as Boolean"}✓
   If CInt(value) > 35 Then Conditional Statement {"IF/End IF"}✓
       MessageBox.Show("Possible highest mark per section is 35.",
                       "Input Data Validation")
       Communication {"MessageBox"}✓
       Return False
       'OR
       ValidateMark = True
   Else
       Return True
       'OR
       ValidateMark = False
       Return Statement {Any ONE"}✓
   End If
End Function
```
(4)

2.7
```
'QUESTION 2.7
Private Sub btnEnter_Click(…) Handles btnEnter.Click
```

2.7.1
```
'QUESTION 2.7.1
 'Check if the limit of the array has been reached
  If intCounterVariable = intNumberOfCandidates Then
  Conditional Statement {"IF/End IF"}✓
     MessageBox.Show("The number of candidates that you can
                     enter has reached its limit", "Limit
                     Exceeded")
     Communication {"MessageBox"}✓
     btnEnter.Enabled = False
     Call ClearInputData()
     Invoke Subprocedure {ClearInputData()}✓
     Exit Sub
     Terminate Statement {"Exit Sub}✓
  End If

Invoke procedure to validate exam number
   ValidateExamNo()
```
(4)

2.7.2
```
'QUESTION 2.7.2
   strArrExamNumber(intCounterVariable) =
CStr(txtExamNumber.Text)
```
Correct array assignment {examination number} (1)

2.7.3
```
'QUESTION 2.7.3
   ' check if section a was answered
   If (CInt(txtSecAMark.Text) > 0) Then
      intArrSecAMark(intCounterVariable) =
CInt(txtSecAMark.Text)
      intNoSectionsAnswered = intNoSectionsAnswered + 1
   End If

   ' check if section B was answered
   If (CInt(txtSecBMark.Text) > 0) Then
```
(4)

2.7.4

```
'QUESTION 2.7.4
```

Increment Number of Sections answered ✓ {Any ONE}

```
      intNoSectionsAnswered = intNoSectionsAnswered + 1

      If intNoSectionsAnswered = 1 Then
         intArrSecAMark(intCounterVariable) = 0
         intArrSecBMark(intCounterVariable) =
CInt(txtSecBMark.Text)
      End If

      If intNoSectionsAnswered = 2 Then
         intArrSecBMark(intCounterVariable) =
CInt(txtSecBMark.Text)
      End If

   End If

  ' check of section C was answered
  If (CInt(txtSecCMark.Text) > 0) Then
        intNoQuestAnswered = intNoQuestAnswered + 1
        If intNoSectionsAnswered = 1 Then
           intArrSecCMark(intCounterVariable) =
                          CInt(txtSecCMark.Text)
        End If

        If intNoSectionsAnswered = 2 Then
           intArrSecCMark(intCounterVariable) =
                          CInt(txtSecCMark.Text)
        End If

        If intNoSectionsAnswered > 2 Then
           intArrSecCMark(intCounterVariable) = 0
        End If
  End If

 ' check of section D was answered
 If (CInt(txtSecDMark.Text) > 0) Then
        intNoSectionsAnswered = intNoSectionsAnswered + 1
        If intNoSectionsAnswered = 1 Then
           intArrSecDMark(intCounterVariable) =
CInt(txtSecDMark.Text)
        End If

        If intNoSectionsAnswered = 2 Then
           intArrSecDMark(intCounterVariable) =
CInt(txtSecDMark.Text)
        End If

        If intNoSectionsAnswered > 2 Then
           intArrSecDMark(intCounterVariable) = 0
        End If
 End If
```

IF statement conditions.
{ If (CInt(txtSecAMark.Text) > 0)}
{ If (CInt(txtSecBMark.Text) > 0)}
{ If (CInt(txtSecCMark.Text) > 0)}
{ If (CInt(txtSecDMark.Text) > 0)}

{Any ONE IF Statement}✓

Array Assignment
{intArrSecAMark(intCounterVariable) = CInt(txtSecAMark.Text)}
{intArrSecBMark(intCounterVariable) = CInt(txtSecBMark.Text)}
{intArrSecCMark(intCounterVariable) = CInt(txtSecCMark.Text)}
{intArrSecDMark(intCounterVariable) = CInt(txtSecDMark.Text)}
{ANY ONE}✓

IF Statement Condition { intNoQuestAnswered = 2 } {Any ONE}✓
IF Statement Condition { intNoQuestAnswered > 2 } {Any ONE}✓

NOTE TO MARKERS: Candidates logic may differ from the above. (1)

2.7.5
```
'QUESTION 2.7.5
 If intNoQuestAnswered > 2 Then
```
Conditional Statement {"If/End If"}✓
```
    strArrMessage(intCounterVariable) = "Too many
questions"
 Else
    strArrMessage(intCounterVariable) = ""
 End If
```
Assign message to array {"both"}✓
```
 'Clear controls to add again
    ClearInputData()
    txtExamNumber.Focus()

 'Enable Display Button
    btnDisplay.Enabled = True

 'Increment the counter
    intCounterVariable += 1
```
(2)

2.7.6
```
'QUESTION 2.7.6
 'Reset the number of questions answered
    intNoQuestAnswered = 0
```
Assign ZERO to number of questions answered✓
```
End Sub
```
(1)

2.8
```
Private Sub btnSearch_Click(…) Handles btnSearch.Click
(…)

'QUESTION 2.8.1
  Dim index As Integer = 0
  For intCounterVariable = 0 To UBound(strArrExamNumber)
    If strArrExamNumber(intCounterVariable) =
txtSearchExaminatioNumber.Text Then
       index = intCounterVariable
       lstDisplayResults.Items.Clear()
       lstDisplayResults.Items.Add("Results: ")

     lstDisplayResults.Items.Add(strArrExamNumber(index).PadRight(18)
        & intArrTotalMarks(index).ToString.PadRight(4) &
        strArrMessage(index).PadLeft(25))
        Exit For
    Else
        lstDisplayResults.Items.Add("Examination number: " &
        txtSearchExaminatioNumber.Text & " not found.
    End If
```

```
Next
```

Loop Statement Condition {0 to upper bounds of array}✓
IF/ELSE Statement Condition✓
Assigning index value { index = intCounterVariable }✓
Formatted Display Output✓ (4)

2.9 Printout of Source Code

2.10

ENGLISH FAL P2

Examination Number	Total Marks	Comment
8306231006089	40	
8706126008084	24	Too many questions

Correct output✓ (1)
[35]

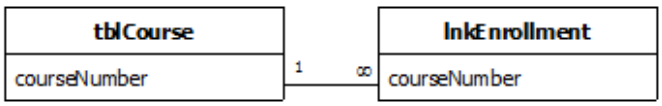
QUESTION 3

3.1 **Columns**

Name	Type	Size
custNumber	Integer	2
courseNumber	Integer	2
dateEnrolled	Date/Time	8
amountPaid	Currency	8

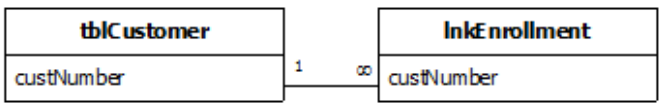
Relationships

tblCourseInkEnrollment



Attributes: Enforced
RelationshipType: One-To-Many

tblCustomerInkEnrollment



Attributes: Enforced
RelationshipType: One-To-Many

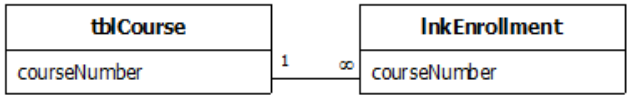
TWO 1:M Relationships✓✓
Enforced Referential Integrity✓

Columns

Name	Type	Size
courseNumber	Integer	2
courseName	Text	50
courseFee	Currency	8

Relationships

tblCourseInkEnrollment



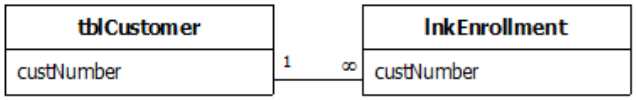
Attributes: Enforced
RelationshipType: One-To-Many

Columns

Name	Type	Size
custNumber	Integer	2
custFirstName	Text	25
custLastName	Text	25
custMobile	Text	10

Relationships

tblCustomerInkEnrollment



Attributes: Enforced
RelationshipType: One-To-Many

Correct table names {All THREE tables}✓
Correct field names {All THREE tables}✓
Correct data Types {All THREE tables}✓
Correct size {All THREE tables}✓ (7)

3.2 3.2.1

custNumber	custFirstName	custLastName	cellPhone
100	Maredi	MOLOTO	0610025540
101	Shane	SMITH	0832349908
102	Meriam	HELEN-AIDE	0726560885
104	Kingsley	KOLOBE	0731123349
105	Anna-Merie	VAN LINDEN	0842007783

ALL correct records✓

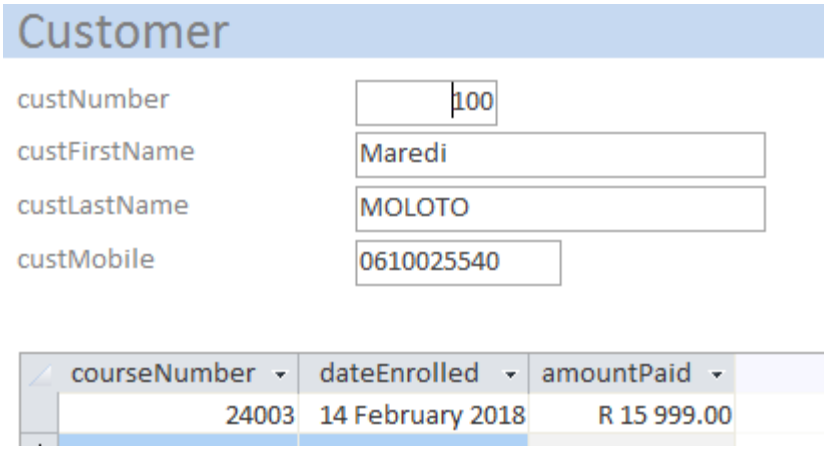
3.2.2

courseNumber	courseName	courseFee
24002	ICT in Education	R 12 500,00
24003	C++ Programming	R 14 000,00
24004	Web design	R 9 200,00
24005	Networking fundamentals	R 5 500,00
24006	Vb.net Programming	R 10 500,00
24007	Introduction to Databases	R 6 500,00

ALL correct records ✓

(2 × 1) (2)

3.3 3.3.1



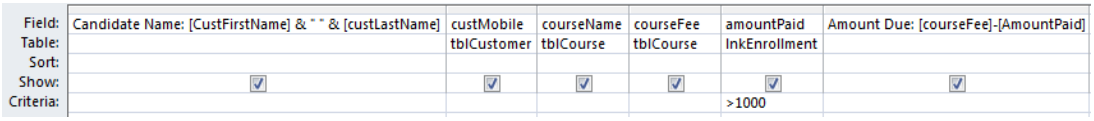
Main form with sub form ✓
Formatted label {"Customer"} ✓ (1)

3.3.2

CustNumber	CourseNumber	AmountPaid
100	24002	R 2 400,00
100	24004	R 1 500,00
101	24003	R 1 500,00
102	24002	R 1 200,00
104	24002	R 500,00
105	24003	R 1 200,00

ALL correct records✓ (1)

3.4 3.4.1–3.4.5



ALL correct fields selected✓ (1)
Calculated Field {"Candidate Name"}✓ (1)
Calculated Field {"Amount Due"}✓✓ (2)
Filtering criteria✓ (1)
Design view printout✓ (1)

3.4.6

Candidate Name	CellPhone	Course Name	AmountPaid	Course Fee	Amount Due
Maredi MOLOTO	0610025540	ICT in Education	R 2 400.00	R 12 500.00	R 10 10
Maredi MOLOTO	0610025540	Web design	R 1 500.00	R 9 200.00	R 7 70
Shane SMITH	0832349908	C++ Programming	R 1 500.00	R 14 000.00	R 12 50
Meriam HELEN-AIDE	0726560885	ICT in Education	R 1 200.00	R 12 500.00	R 11 30
Anna-Merie VAN LINDEN	0842007783	C++ Programming	R 1 200.00	R 14 000.00	R 12 80

ALL correct records✓ (1)

3.5

The Course List - EXAMINATION NUMBER

Course Name	Candidate Name	CellPhone	AmountPaid	Course Fee	Amount Due
C++ Programming					
	Anna-Merie VAN LINDEN	0842007783	R 1 200.00	R 14 000.00	R 12 800.00
	Shane SMITH	0832349908	R 1 500.00	R 14 000.00	R 12 500.00
			R 2 700.00	R 28 000.00	R 25 300.00
ICT in Education					
	Kingsley KOLOBE	0731123349	R 500.00	R 12 500.00	R 12 000.00
	Meriam HELEN-AIDE	0726560885	R 1 200.00	R 12 500.00	R 11 300.00
	Maredi MOLOTO	0610025540	R 2 400.00	R 12 500.00	R 10 100.00
			R 4 100.00	R 37 500.00	R 33 400.00
Web design					
	Maredi MOLOTO	0610025540	R 1 500.00	R 9 200.00	R 7 700.00
			R 1 500.00	R 9 200.00	R 7 700.00
			R 8 300.00	R 74 700.00	R 66 400.00

All correct fields✓ (1)
Correct Grouping level {"Course Name"}✓ (1)
Totals, Subtotals and Grand Totalled report✓✓✓ (3)
Report title✓ (1)
Report printout✓ (1)
[25]

TOTAL: 100

Copyright reserved