higher education & training

Department:
Higher Education and Training
REPUBLIC OF SOUTH AFRICA

MARKING GUIDELINE

NATIONAL CERTIFICATE (VOCATIONAL)

PRINCIPLES OF COMPUTER PROGRAMMING
NQF LEVEL 3
(Second paper)

28 November 2023

This marking guideline consists of 9 pages.

Copyright reserved

INSTRUCTIONS TO MARKERS

1. Mark all the questions, using RED ink only.

2. Carefully read all the answers provided in the source code.

3. Make the ticks neatly and initial each page.

4. Markers are required to work out the question paper before the marking guideline is discussed.

5. The marking guideline is to be used as the basis for the marking session. All markers are required to attend a rigorous standardisation meeting to ensure that these guidelines are interpreted consistently and applied in the marking of candidates' work.

6. Note that candidates who provide an alternate correct solution to that given as a solution in the marking guideline, will be given full credit for the relevant solution, unless the specific instructions in the paper were not followed or the requirements of the question were not met.

7. The candidate's total mark is written on the EXAMINATION FOLDER cover next to the corresponding question number.

8. If a candidate handed in a PRINTOUT, but achieved 0 for the question, then 0 (zero) is written on the cover next to the corresponding question number.

9. If a candidate did not hand in a PRINTOUT, a line is drawn through the block next to the corresponding question number.

10. If a candidate handed in a PRINTOUT without a printed EXAMINATION NUMBER, the question must be marked and reported as an irregularity.

11. All irregularities should be marked and marks transferred to the cover page. The word 'Irregularity' must be written next to the percentage and an irregularity report must be completed.

13. If a candidate handed in duplicate printouts, a line must be drawn across the duplicate with the word 'Duplicate' written across it.

Copyright reserved

MARKING GRID

QUESTION 1:

QUESTION NUMBER	DESCRIPTION	MAX MARKS
1.1.1	`Dim age As Integer`✓ `Dim dependents As Integer` `Dim medpayment As Double`✓ **[ANY 2 X 1]**	(2)

1.1.2	`If txtAge.Text = "" Then`✓
	`  MessageBox.Show("Please enter your Age")`✓
	`  Return`
	`ElseIf txtNoDependants.Text = "" Then`✓
	`  MessageBox.Show("Please enter your dependants (0 if none)")` ✓
	`  Return`
	`End If`
	
	ANY PROGRAMMATICALLY CORRECT AND WORKING VALIDATION TEST IS ACCEPTABLE
		(4)

1.1.3	`age = Convert.ToInt32(txtAge.Text)` ✓ `dependents = Convert.ToInt32(txtNoDependants.Text)` ✓	(2)

1.1.4	`If (radGold.Checked) Then`✓
	`  If (age >= 35) Then`✓
	`  medpayment = 1800 + (1800 * 0.45) * dependents`✓
	`  MessageBox.Show("Your monthly payment for the Gold pacakge is: " & medpayment.ToString("C2"))`
	`  Else`
	`  medpayment = 1500 + (1500 * 0.45) * dependents`✓
	`  MessageBox.Show("Your monthly payment for the Gold pacakge is: " & medpayment.ToString("C2"))`
	`  End If`
	`ElseIf (radSilver.Checked) Then`✓
	`  If (age >= 35) Then`✓
	`  medpayment = 1400 + (1400 * 0.35) * dependents`✓
	`  MessageBox.Show("Your monthly payment for the Silver Package is: " & medpayment.ToString("C2"))`
	`  Else`
	`  medpayment = 1100 + (1100 * 0.35) * dependents`✓
	`  MessageBox.Show("Your monthly payment for the Silver Package is: " & medpayment.ToString("C2"))`
	`  End If`
	`ElseIf (radCopper.Checked) Then`✓
	`  If (age >= 35) Then`✓
	`  medpayment = 900 + (900 * 0.25) * dependents`✓
	`  MessageBox.Show("Your monthly payment for the Copper Package is: " & medpayment.ToString("C2"))`
	`  Else`
	`  medpayment = 750 + (750 * 0.25) * dependents`✓
		(12)

Copyright reserved

	MessageBox.Show("Your monthly payment for the Copper Package is: " & medpayment.ToString("C2")) End If End If **ANY OTHER WORKING VARIATION OF A "NESTED IF" SOLUTION IS ACCEPTABLE**	
1.1.5	1 Mark for each screenshot with correct output: SILVER: R2640,00 pm✓ GOLD: R1800,00 pm ✓	(2)
1.1.6	Source code printout (no mark)	(0)
		[22]

QUESTION 2:

QUESTION NUMBER	DESCRIPTION	MAX MARKS
2.1	txtAccBalance.Text = ""✓ lstInstallments.Items.Clear()✓	(2)
2.2.1	Dim period As Integer✓ Dim balance As Double Dim installment As Double✓ **[ANY 2 X 1]**	(2)
2.2.2	period = Convert.ToInt32(cmbPaymentPeriod.Text) ✓ balance = Convert.ToDouble(txtAccBalance.Text) ✓	(2)
2.2.3	Select Case period ✓ Case 3✓ balance = balance + (0.12 * balance) ✓ installment = balance / period For r = 1 To period✓ lstInstallments.Items.Add("Installment #" & r.ToString() & ": " & installment.ToString("C2"))✓	

	Next lstInstallments.Items.Add("TOTAL PAID : " & balance.ToString("C2"))✓ Case 6 balance = balance + (0.18 * balance) ✓ installment = balance / period For r = 1 To period✓ lstInstallments.Items.Add("Installment #" & r.ToString() & ": " & installment.ToString("C2"))✓ Next lstInstallments.Items.Add("TOTAL PAID : " & balance.ToString("C2"))✓ Case 9 balance = balance + (0.26 * balance) ✓ installment = balance / period For r = 1 To period✓ lstInstallments.Items.Add("Installment #" & r.ToString() & ": " & installment.ToString("C2"))✓ Next lstInstallments.Items.Add("TOTAL PAID : " & balance.ToString("C2"))✓ End Select **ANY OTHER WORKING VARIATION IN TERMS OF CALCULATING THE INSTALMENT AND DISPLAYING THE REQUIRED RESULTS IS ACCEPTABLE, BUT IT MUST USE A "SELECT CASE" SELECTION STRUCTURE.**	(14)
2.2.4	1 Mark for each screenshot with correct output: **R2200.00 over 6 months:** instalment R432.67 total paid: R2596.00✓ **R1380.00 over 9 months:** instalment R193.20 total paid: R1738.00✓	(2)
2.2.5	Source code printout (no mark)	(0)
		[22]

QUESTION 3:

QUESTION NUMBER	DESCRIPTION	MAX MARKS
3.1	Dim arrKLValues(4) As Integer✓	(1)
3.2	Dim daycnter As Integer✓ Dim totKiloLitres As Integer Dim aveKL As Double Dim totCost As Double✓ Dim aveCost As Double	

	Dim highestKL As Integer Dim highestDay As Double Dim dayname As String **ANY 2 X 1**	(2)
3.3.1	If txtDayKL.Text = "" Then✓ MessageBox.Show("Please provide a value in Kilo Litres")✓ Return End If	(2)
3.3.2	If daycnter > 5 Then✓ MessageBox.Show("List is full - only 5 values are allowed")✓ Return End If **ANY OTHER PROGRAMMATICALLY CORRECT VALIDATION TEST FOR THE UPPERBOUND OF THE LIST IS ACCEPTABLE**	(2)
3.3.3	lstAnalysis.Items.Add("#" & daycnter.ToString() & " : water usage = " & arrKLValues(daycnter - 1).ToString() & " KL")✓ txtDayKL.Text = ""✓ txtDayKL.Focus()✓	(3)

3.4.1	If (daycnter <> 5) Then ✓
	MessageBox.Show("Valid values are required to compute analysis")✓
	Return
	End If
	
	ANY OTHER PROGRAMMATICALLY CORRECT VALIDATION TEST FOR THE UPPERBOUND OF THE LIST IS ACCEPTABLE
		(2)

3.4.2	For i = 0 To 4✓ If highestKL < arrKLValues(i) Then✓ highestKL = arrKLValues(i) ✓ Select Case i✓ Case 0 dayname = "Monday" Case 1 dayname = "Tuesday" Case 2 dayname = "Wednesday" Case 3 dayname = "Thursday" Case 4 dayname = "Friday"✓ End Select End If totKiloLitres = totKiloLitres + arrKLValues(i) ✓ Next	(6)

	ANY OTHER WORKING VARIATION IN TERMS OF CALCULATING THE REQUIRED VALUES IS ACCEPTABLE	
3.4.3	`totCost = 23.75 * totKiloLitres` ✓ `aveCost = totCost / 5` ✓ `aveKL = totKiloLitres / 5` ✓ `highestDay = highestKL * 23.75` ✓	(4)

3.4.4	`lstAnalysis.Items.Add("Total Kilo Litres used: " & totKiloLitres.ToString())` ✓
	`lstAnalysis.Items.Add("Total Cost Incurred: " & totCost.ToString("C2"))` ✓
	`lstAnalysis.Items.Add("Average Kilo Litre Usage: " & aveKL.ToString())`
	`lstAnalysis.Items.Add("Average Cost Incurred: " & aveCost.ToString("C2"))` ✓
	`lstAnalysis.Items.Add("Highest Daily KL Consumption: " & highestKL.ToString())` ✓
	`lstAnalysis.Items.Add("Highest Daily KL Cost : " & highestDay.ToString("C2"))`
	`lstAnalysis.Items.Add("Most Expensive Day (Usage): " & dayname)`
	ANY 4 X 1
		(4)

3.4.5	Source code printout (no mark)	(0)
		[26]

QUESTION 4:

QUESTION NUMBER	DESCRIPTION	MAX MARKS

4.1	`If System.IO.File.Exists("students.csv") Then` ✓
	`  lblFileStatus.Text = "File Exists: READY to save to file"` ✓
	`Else`
	`  lblFileStatus.Text = "File will be created when SUBMIT button is clicked"` ✓
	`End If`
		(3)

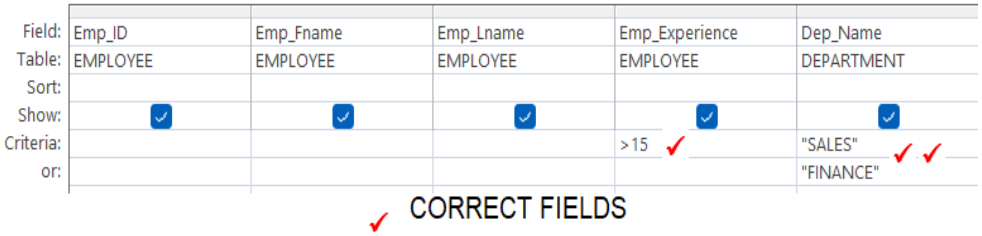
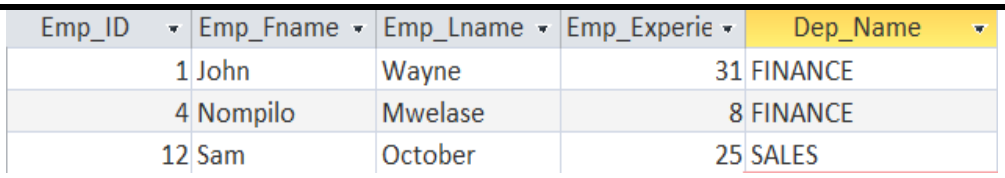
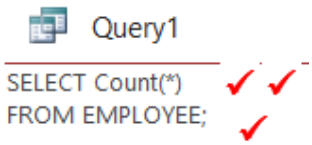
4.2	`Dim filename As String = "students.csv"` ✓ `Dim recWriter As New System.IO.StreamWriter(filename, True)` ✓ `Dim studRecord As String` `Try` ✓ `studRecord = txtRecID.Text & "," & txtFirstName.Text & "," & txtSurname.Text & "," & txtTelephone.Text & "," & txtMobileNo.Text & "," & txtEmail.Text & "," & vbCrLf` ✓	(9)

```
recWriter.WriteLine(studRecord) ✓
recWriter.Close()✓

txtFirstName.Text = ""
txtSurname.Text = ""
txtRecID.Text = ""
txtMobileNo.Text = ""✓
txtTelephone.Text = ""
txtEmail.Text = ""
txtRecID.Focus()

'—INFORM USER THAT RECORD WAS ADDED
MessageBox.Show("SUCCESS: Student Record Saved to File")✓

Catch ex As Exception
'-- INFORM USER RECORD COULD NOT BE SAVED
MessageBox.Show("ERROR: Student Record was NOT Saved")✓
End Try
```

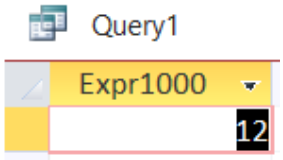
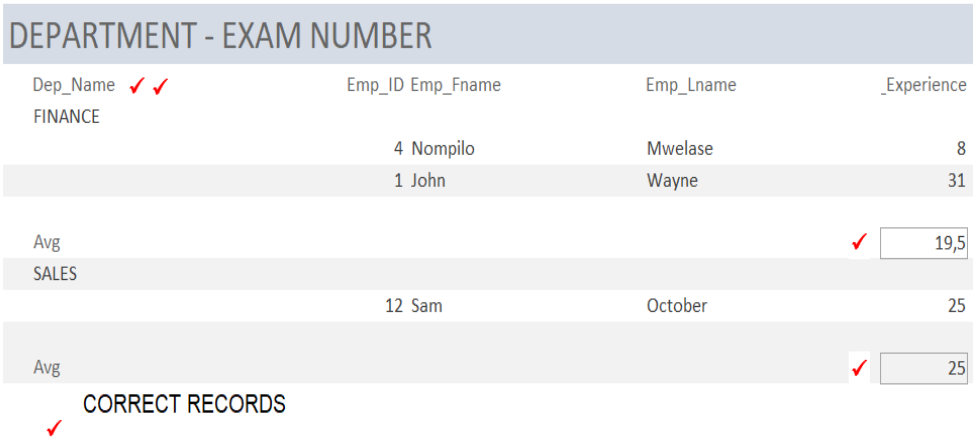
ANY OTHER WORKING VARIATION OF THE ABOVE SOLUTION IS ACCEPTABLE

4.3	Source code printout (no mark)	**(0)**
		[12]

QUESTION 5:

QUESTION NUMBER	DESCRIPTION	MAX MARKS
5.1	Printout of table structures showing “enforced” relationship	(2)
5.2		(4)
5.3		(2)
5.4		(3)

Emp_ID	Emp_Fname	Emp_Lname	Emp_Experie	Dep_Name
1	John	Wayne	31	FINANCE
4	Nompilo	Mwelase	8	FINANCE
12	Sam	October	25	SALES

5.5		(2)
5.6		(5)
		[18]

TOTAL [100]