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

11 November 2022

This marking guideline consists of 11 pages.

Copyright reserved

1. Markers are required to work out the answers to the question paper before the marking guideline discussion.

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

3. Note that candidates who provide an alternate correct solution to that given as example of 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.

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

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

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

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

8. All irregularities are marked, and marks are transferred to the cover page. The word, 'Irregularity,' is written next to the percentage and an irregularity report is completed.

9. If a candidate handed in duplicate PRINTOUTS, a line must be drawn across the duplicate with a notation 'Duplicate' written across it.

Copyright reserved

QUESTION 1: MARKING GRID: GENERAL PROGRAMMING SKILLS

QUESTION	NUMBER	DESCRIPTION	MARK	MAX MARKS
1.1		Button – Question 1_1		
	1.1.1	If ID number text box is empty. If txtIdNumber.Text = Nothing OR If txtIdNumber.Text = ""	✓	
		Change the back colour of text box txtIdNumber.BackColor = Color.Red	✓	
		Move cursor to ID Number text box. txtIdNumber.Focus()	✓	
		Terminate the procedure. Exit Sub	✓	(4)
	1.1.2	The identity number should consist of numbers only. (Mark positive and/or negative logic.) If IsNumeric(txtIdNumber.Text) = True OR If Not Long.TryParse(txtIdNumber.Text, 0)	✓	
		The identity number cannot be shorter or longer than thirteen (13) numbers. (Mark positive and/or negative logic.) If txtIdNumber.TextLength <> 13	✓	
		Assign the identity number to the provided global variable. strIdentityNumber = txtIdNumber.Text	✓	
		Displaying a pop-up dialog box, moving the cursor to the ID number text box and terminating the procedure on invalid entry. MessageBox.Show("…") txtIdNumber.Focus() Exit Sub (All three)	✓	(4)
	1.1.3	If first name and surname consist of two (2) or more characters: If txtFirstName.TextLength <= 1 If txtSurName.TextLength <= 1	✓✓	
		Displaying a pop-up dialog box to indicate the specific error: MessageBox.Show("…") txtIdNumber.Focus() Exit Sub (All three)	✓	(3)
	1.1.4	Concatenate name and surname and display in the label. lblDisplayCustInfo.Text = "Dear " & txtFirstName.Text & " " & txtSurname.Text & vbCrLf & "Welcome to Mzansi Superstore's Loyalty Programme"	✓✓	(2)

QUESTION	NUMBER	DESCRIPTION	MARK	MAX MARKS
1.2		Radio buttons – CheckChanged Event Procedure If stand-alone programme selected points = 1000 If radStandAlone.Checked = True intPoints = 1000 } (both) If multi-partner programme selected points = 500. If radMultiPartner.Checked = True intPoints = 500 } (both) **NOTE:** The above code must be in the correct event procedure. Penalise only one (1) mark if the code is written in another event procedure.	✓ ✓	(2)
1.3		Button – Question 1_3 Calculate total loyalty points. Display a message indicating the total loyalty points lblDisplayPoints.Text = "Dear customer, you have accumulated " & ((CInt(txtTotalAmount.Text) / 5) + intPoints).ToString & " points."	✓✓ ✓✓	(4)
1.4		Button – Question 1_4 If generate membership number check box is selected. If chkLoyaltyCard.Checked = True If standalone membership is selected: If radStandAlone.Checked = True Generate membership number: strMembershipNumber = "S" & strIdentityNumber.Substring(0, 6) & Now.ToString("dd/MM/yyyy") If multi partner membership is selected: If radMultiPartner.Checked = True Then Generate membership number: strMembershipNumber = "M" & strIdentityNumber.Substring(0, 6) & Now.ToString("dd/MM/yyyy") Display output in label. lblDisplayMembershipNumber.Text = strMembershipNumber	✓ ✓ ✓ ✓ ✓ ✓	(6)
				[25]

QUESTION 2: MODULAR PROGRAMMING (ARRAYS AND PROCEDURES)

QUESTION	NUMBER	DESCRIPTION	MARK	MAX MARKS

2.1		Create a loop counter variable.
		`Dim intCount As Integer = 0`
		
		Create three (3) parallel, single dimension arrays.
		`Dim intArrKmsTravelled(6) As Integer`
		`Dim intArrLtsUsed(6) As Integer`
		`Dim decArrConsumption(6) As Decimal`
			✓
			
			✓✓✓
				(4)

2.2		Function procedure (CalcConsumption)
		
		Function keyword.
		Correct parameter type. (ByVal)
		Correct input parameters.
		Correct parameter datatype. (Integer)
		Function return type. (Decimal)
		Return Calculation.
		
		`Public Function CalcConsumption(ByVal KmsTravelled As Integer, ByVal Fuel As Integer) As Decimal`
		
		`Return CDec((Fuel / KmsTravelled) * 100)`
			
			✓
			✓
			✓
			✓
			✓
			✓✓
				(7)

2.3	2.3.1	Button – Question 2_3
		
		Assign text boxes to arrays.
		Datatype conversion function. (Cint)
		
		`intArrKmsTravelled(intCount) = CInt(txtDistance.Text)`
		`intArrLtsUsed(intCount) = CInt(txtFuel.Text)`
			
			✓✓
			✓
				(3)

	2.3.2	Increment loop counter `intCount += 1 OR intCount = intCount + 1`	✓	(1)

	2.3.3 (a)	If loop counter / upper bound of array reached.
		`If intCount <= 6 OR If UBound(…) <= 6`
		
		Terminate procedure
		`Exit Sub`
			✓
			
			✓
				(2)

	2.3.3 (b)	If FALSE Else Disable group box. grpInput.Enabled = False Clear list box. lstAnalysis.Items.Clear() Draw correct headings. lstAnalysis.Items.Add("No" & "Distance".PadLeft(10) & "Fuel_Used".PadLeft(12) & "Consumption".PadLeft(16)) lstAnalysis.Items.Add("(l/100km)".PadLeft(40)) lstAnalysis.Items.Add("----------") Iterate through Distance Travelled array to calculate fuel consumption. For i = 0 To UBound(intArrKmsTravelled) decArrConsumption(i) = CalcConsumption(intArrKmsTravelled(i), intArrLtsUsed(i)) lstAnalysis.Items.Add((i + 1) & intArrKmsTravelled(i).ToString.PadLeft(10) & intArrLtsUsed(i).ToString.PadLeft(12) & decArrConsumption(i).ToString("N2").PadLeft(16)) Declare variables for totals. Dim intTotalDistance As Integer Dim intTotalFuel As Integer Dim decTotalFuelConsumption As Decimal Iterate through Distance Travelled array to calculate totals. For i = 0 To UBound(intArrKmsTravelled) intTotalDistance += intArrKmsTravelled(i) intTotalFuel += intArrLtsUsed(i) decTotalFuelConsumption += decArrConsumption(i) Display totals. lstAnalysis.Items.Add("Total Distance: " & intTotalDistance & " kms travelled") lstAnalysis.Items.Add("Total Fuel Used: " & intTotalFuel & " litres used")	✓ ✓ ✓ ✓✓ ✓✓ ✓✓ ✓ ✓ ✓ ✓	

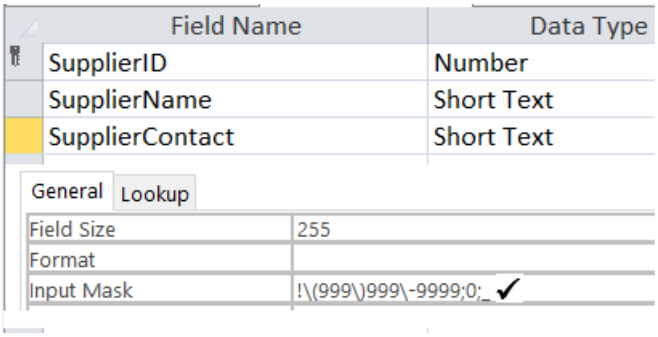
		Calculate and display averages. Correct number formatting (“N2”) lstAnalysis.Items.Add("Average Distance: " & (intTotalDistance / (UBound(intArrKmsTravelled) + 1)).ToString("N2") & " kms.") lstAnalysis.Items.Add("Average Fuel Used: " & (intTotalFuel / (UBound(intArrLtsUsed) + 1)).ToString("N2") & " litres.") lstAnalysis.Items.Add("Average Fuel Consumption: " & (decTotalFuelConsumption / (UBound(intArrLtsUsed) + 1)).ToString("N2") & " l/100km")	✓	(18)
				[35]

QUESTION 3: FILE AND ERROR HANDLING

QUESTION	NUMBER	DESCRIPTION	MARK	MAX MARKS
3.1	3.1.1	Form Load procedure (frmQuestion3_Load) Exception Handler (Try,Catch) Try Stream reader object Dim reader As StreamReader 'reader object Line counter variable Dim intLineCount As Integer = 0 'line counter Load file into reader reader = New StreamReader("dictionary.txt") Prevent EOF While Not reader.EndOfStream Readlines lstWords.Items.Add(reader.ReadLine) Increment line counter intLineCount += 1 Close reader reader.Close() Display output lblLineCount.Text = intLineCount & " lines of text" Catch ex As Exception 'Display any possible error message MessageBox.Show("The file cannot be loaded " & vbCrLf & ex.Message)	✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	(11)
3.2	3.2.1	Find word in the list box and assign the result to a new integer variable. Dim index As Integer = lstWords.FindString(txtLetter.Text)	✓✓	(2)
	3.2.2	Highlight search word lstWords.SetSelected(index, True)	✓✓	(2)
				[15]

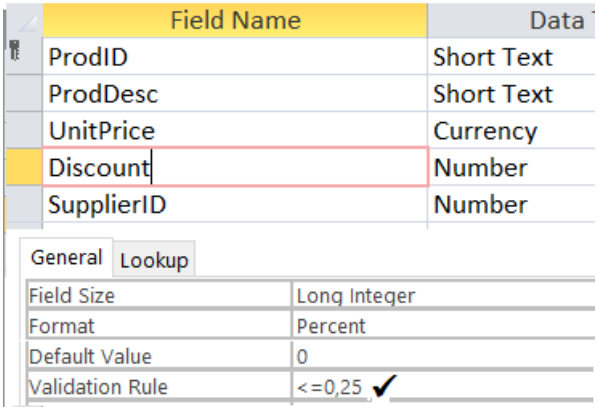
QUESTION 4: DATABASE

4.1 4.1.4



Input mask (1)
Printout (1) (2)

4.1.5



Correct validation rule (1)
Printout (1)

NOTE: The validation rule must be 0,25 and not 25. (2)

4.1.6

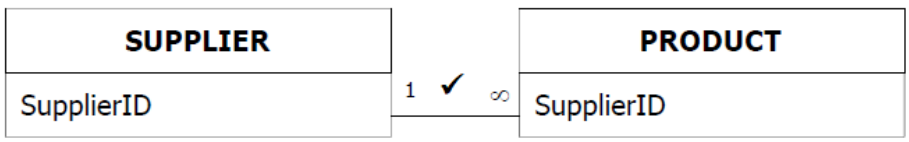
Table: PRODUCT Page: 1

Columns

Name	Type	Size
ProdID	Short Text	8
ProdDesc	Short Text	40
UnitPrice ✓	Currency ✓	8 ✓
Discount	Long Integer	4
SupplierID	Long Integer	4

Relationships

SUPPLIERPRODUCT



Attributes: Enforced
RelationshipType: One-To-Many

Correct field names (1) (4)

Correct data types (1)
Correct field size (1)
Correct relationship type (1)

4.1.6

Table: SUPPLIER Page: 2

Columns

Name	Type	Size
SupplierID	Long Integer	4
SupplierName ✓	Short Text ✓	22 ✓
SupplierContact	Short Text	255

Relationships

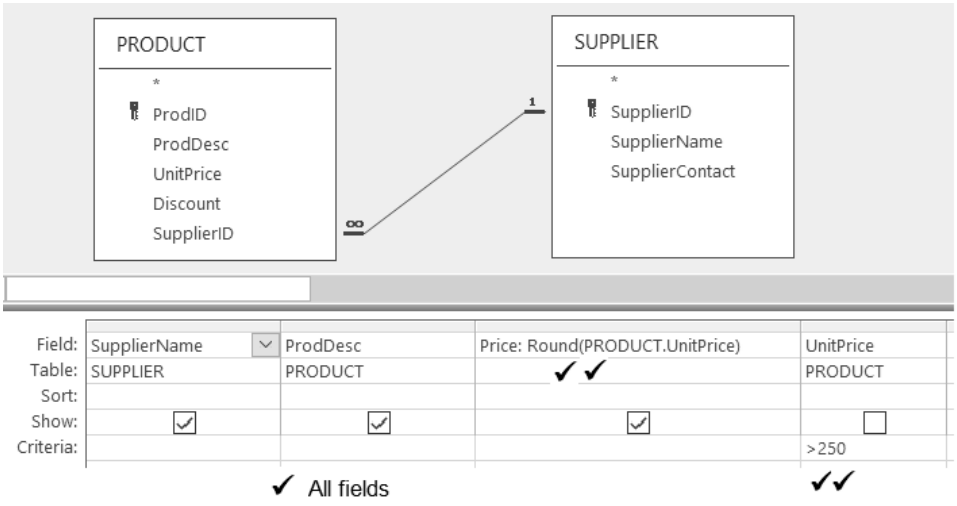
SUPPLIERPRODUCT

SUPPLIER	✓	PRODUCT
SupplierID	1 ∞	SupplierID

Attributes: Enforced
RelationshipType: One-To-Many

Correct field names (1)
Correct data types (1)
Correct field size (1)
Enforce referential integrity (1) (4)

4.3 4.3.1



Display all correct fields (1)
Round formula (2)
Correct criteria (2) (5)

4.3.2

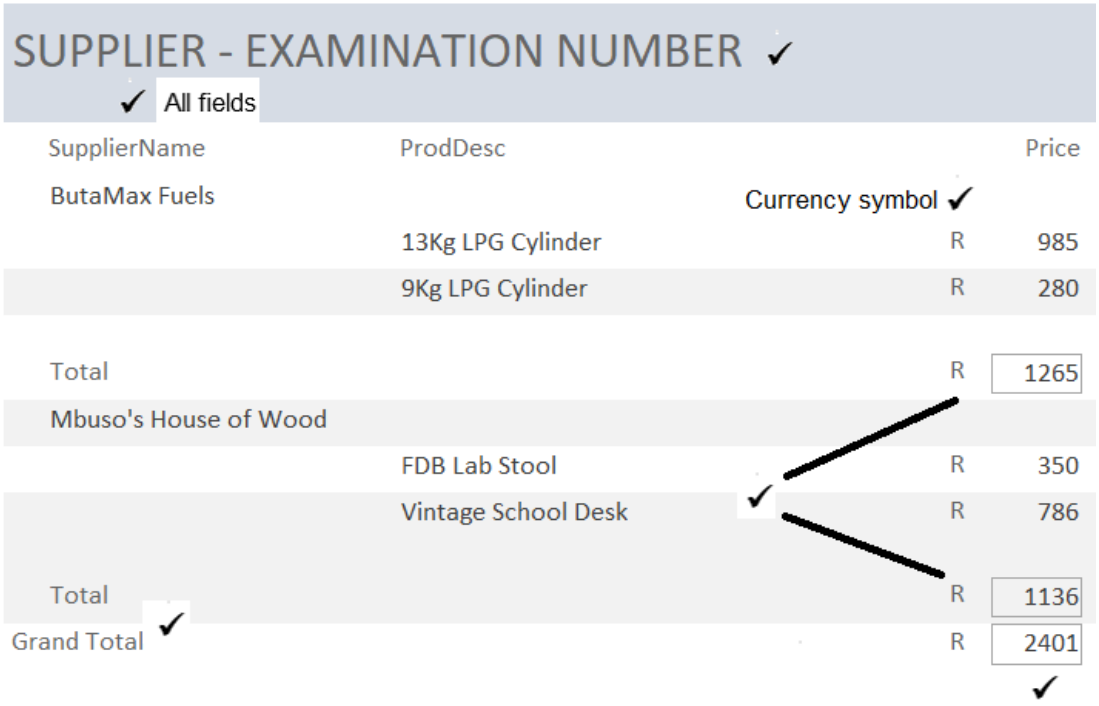
SupplierName	ProdDesc	Price
ButaMax Fuels	9Kg LPG Cylinder	280
Mbuso's House of Wood	Vintage School Desk	786
ButaMax Fuels	13Kg LPG Cylinder	985
Mbuso's House of Wood	FDB Lab Stool	350

✓ All fields

Display all correct fields (1) (2)

Printout (1)

4.4



Display all correct fields	(1)
Examination number in header	(1)
Currency symbol	(1)
Total labels	(1)
Summary totals	(1)
Printout	(1)

(6)
[25]

TOTAL: 100

Copyright reserved