ICT2611

 |

Assignment 4

Student Number: 64482790

2021

Contents

[Chapter 5 - Programming Project 3 - Furniture Order 2](#_Toc79745226)

[Code 2](#_Toc79745227)

[Images 4](#_Toc79745228)

[Chapter 6 - Programming Project 7 – Palindrome 5](#_Toc79745229)

[Code 5](#_Toc79745230)

[Images 6](#_Toc79745231)

[Chapter 7 - Programming Project 2 - ISBN Validator 8](#_Toc79745232)

[Code 8](#_Toc79745233)

[Images 9](#_Toc79745234)

# Chapter 5 - Programming Project 3 - Furniture Order

## Code

Public Class Form1

Private Sub btnProcess\_Click(sender As Object, e As EventArgs) Handles btnProcess.Click

Dim cusName, address, city As String

Dim numChair, numSofa, price As Double

cusName = txtName.Text

address = txtAddress.Text

city = txtCity.Text

numChair = CDbl(txtNumChair.Text)

numSofa = CDbl(txtNumSofa.Text)

'adds text into rich edit

edtDisplay.AppendText("Invoice Number: " & OrderNum(cusName, city) & vbCrLf & vbCrLf & vbCrLf)

edtDisplay.AppendText("Name: " & NameCorr(cusName) & vbCrLf)

edtDisplay.AppendText("Address: " & address & vbCrLf)

edtDisplay.AppendText("City: " & city & vbCrLf & vbCrLf & vbCrLf)

edtDisplay.AppendText("Number of Chairs: " & numChair & vbCrLf)

edtDisplay.AppendText("Number of Sofas: " & numSofa & vbCrLf & vbCrLf & vbCrLf)

price = numChair \* 350 + numSofa \* 925

edtDisplay.AppendText(" Price: " & price.ToString("C") & vbCrLf)

edtDisplay.AppendText(" Sales Tax: " & (price \* 0.05).ToString("C") & vbCrLf)

edtDisplay.AppendText(" ----------------" & vbCrLf)

edtDisplay.AppendText(" Total Price: " & (price + price \* 0.05).ToString("C") & vbCrLf)

End Sub

Function OrderNum(cusName As String, city As String) As String 'function to create order number

Dim invoiceLet, invoiceNum As String

Dim pos As Integer

invoiceLet = cusName.Substring(0, 2) 'gets the first two letters of surname

invoiceLet = invoiceLet.ToUpper

pos = city.Length

invoiceNum = city.Substring(pos - 4) 'gets the last 4 digits of the zip code

Return (invoiceLet & invoiceNum)

End Function

Function NameCorr(cusName As String) As String 'function to convert name

Dim name As String

Dim pos As Integer

pos = cusName.IndexOf(" ") 'finds the space between the name and surname

name = cusName.Substring(pos + 1) 'gets the first name

pos = cusName.Length - name.Length - 2

name = name & " " & cusName.Substring(0, pos) 'adds last name

Return name

End Function

Private Sub btnClear\_Click(sender As Object, e As EventArgs) Handles btnClear.Click 'function to clear form

'clears all fields as well as rich edit

txtName.Clear()

txtAddress.Clear()

txtCity.Clear()

txtNumChair.Clear()

txtNumSofa.Clear()

edtDisplay.Clear()

End Sub

Private Sub btnClose\_Click(sender As Object, e As EventArgs) Handles btnClose.Click 'function to close application

'closes the application

Me.Close()

End Sub

End Class

## Images

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generated

# Chapter 6 - Programming Project 7 – Palindrome

## Code

Public Class Form1

Private Sub btnDeterm\_Click(sender As Object, e As EventArgs) Handles btnDeterm.Click

Dim trimChar As Char() = {""}

Dim word, wordFormat As String

word = txtWord.Text

wordFormat = ""

'gets only the letters out of the string

For i As Integer = 0 To word.Length - 1

If word.Chars(i) Like "[A-Za-z]" Then

wordFormat &= word.Chars(i)

End If

Next

If isPalindrome(wordFormat.ToLower) = True Then

txtOut.Visible = True

txtOut.Text = "Yes"

Else

txtOut.Visible = True

txtOut.Text = "No"

End If

End Sub

Function isPalindrome(wordFormat As String) As Boolean 'function to determine if palindrome

Dim output As Boolean

Dim d As Integer

output = True

d = wordFormat.Length - 1

'checks if the letters are the same

For i As Integer = 0 To (wordFormat.Length - 1) / 2

If wordFormat.Chars(i) <> wordFormat.Chars(d) Then

output = False

End If

d -= 1

Next

Return output

End Function

End Class

## Images

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

# Chapter 7 - Programming Project 2 - ISBN Validator

## Code

Public Class Form1

Private Sub Validate\_Click(sender As Object, e As EventArgs) Handles Validate.Click

Dim ISBN, format As String

Dim numISBN(9) As String

Dim d, sum As Integer

ISBN = txtISBN.Text

format = ""

If ISBN.Chars(ISBN.Length - 1) = "X" Then 'determines if the last digit is an x

numISBN(9) = 10

Else

numISBN(9) = CInt(ISBN.Substring(ISBN.Length - 1))

End If

For i As Integer = 0 To (ISBN.Length - 1) 'formats the string without the -

If ISBN.Chars(i) Like "[0-9]" Then

format &= ISBN.Chars(i)

End If

Next

For i As Integer = 0 To 8 'adds each digit into array

numISBN(i) = CInt(format.Substring(i, 1))

Next

d = 10

sum = 0

For i As Integer = 0 To 9 'sums the values

sum += numISBN(i) \* d

d -= 1

Next

If sum Mod 11 = 0 Then

txtValid.Text = "Yes"

Else

txtValid.Text = "No"

End If

End Sub

End Class

## Images

Graphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generated