# Practical 6

Program to print factorial series

Module exp6\_1

Sub Main()

Dim f, n, n2 As Integer

Console.WriteLine("Enter the number to calculate the factorial of: ") n = 1 n2 = 1

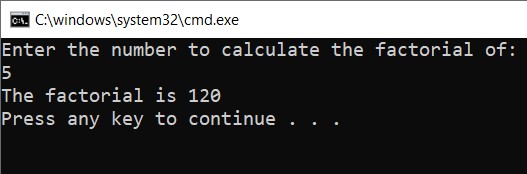
f = Console.ReadLine() While n <= f n2 = n2 \* n n = n + 1

End While

Console.WriteLine("The factorial is " & n2)

End Sub

End Module



Print Fibonacci series

Module exp6\_2

Sub main()

Dim a As Integer = 0

Dim b As Integer = 1

Dim temp As Integer

Console.WriteLine("Fibonacci Series:")

Console.WriteLine(a)

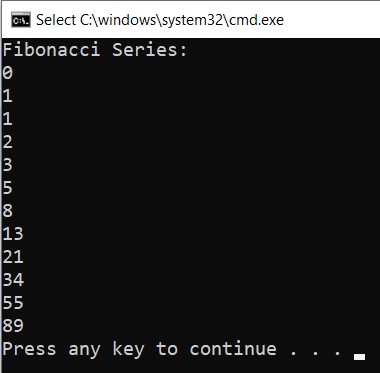
Console.WriteLine(b) For i = 0 To 10 - 1 temp = a a = b b = temp + b

Console.WriteLine(b)

Next

End Sub

End Module



Print Prime series..

Module exp6\_4

Sub main()

Dim j As Integer

Dim i As Double Dim t As Boolean j = 2

Console.Write("Enter a Number: ") i = Console.ReadLine()

t = True While j < i

If i Mod j = 0 Then t = False End If j = j + 1 End While

If t Then

Console.WriteLine(i & " is a prime number")

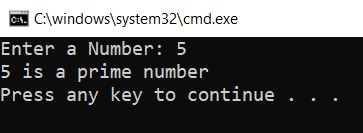
Else

Console.WriteLine(i & " is not a prime number")

End If

End Sub

End Module



Print even and odd numbers between 1 to 50

Module exp6\_3

Sub main()

Dim n1, n2 As Integer n1 = 1 n2 = 1

Console.WriteLine("Even series:")

While n1 <= 50

If n1 Mod 2 = 0 Then

Console.Write(n1 & " ") End If n1 = n1 + 1

End While

Console.WriteLine(vbCrLf & "Odd Series:")

While n2 <= 50

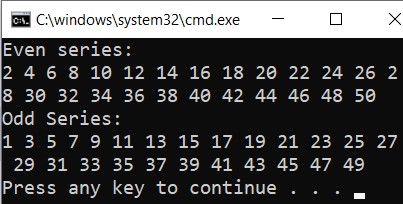
If n2 Mod 2 = 1 Then

Console.Write(n2 & " ") End If n2 = n2 + 1 End While

Console.WriteLine()

End Sub

End Module



# Practical 7

Sum and average of 5 subjects using For loop

Module Module1

Sub Main()

Dim i, n, s As Integer

Dim a As Double

Console.WriteLine("Enter marks for 5 subjects") For i = 1 To 5 n = Console.ReadLine()

s = s + n

Next

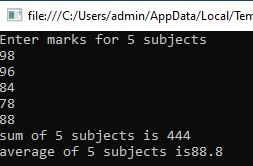
Console.WriteLine("sum of 5 subjects is " & s)

a = s / 5

Console.WriteLine("average of 5 subjects is" & a) Console.ReadLine()

End Sub

End Module



Print reverse of 1 to 10 using For loop

Module Module2

Sub main()

Dim i As Integer

For i = 10 To 1 Step -1

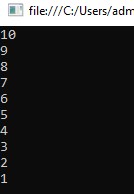
Console.WriteLine(i)

Next

Console.ReadLine()

End Sub

End Module



Print multiplication table of a number

Module Module3

Sub main()

Dim i, n As Integer

Console.WriteLine("enter any number")

n = Console.ReadLine() For i = 1 To 10 Step 1

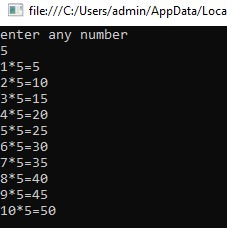
Console.WriteLine(i & "\*" & n & "=" & (n \* i))

Next

Console.ReadLine()

End Sub

End Module



# Practical 24

Design the windows application that will display the content of a table in

MS-Access database on DataGrid control using data adapter

Imports System.Data.OleDb

Public Class Form1

Dim cmd As OleDbCommand = New OleDbCommand

Dim reader As OleDbDataReader

Dim conn As OleDbConnection = New OleDbConnection

Dim adpt As OleDbDataAdapter = New OleDbDataAdapter

Dim ds As New DataSet

Private Sub Form3\_Load(sender As Object, e As EventArgs) Handles MyBase.Load conn = New

OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Users\siddi\OneDrive\Documents\Database2.mdb;") conn.Open()

cmd = New OleDbCommand("select \* from Visual", conn) adpt = New OleDbDataAdapter(cmd)

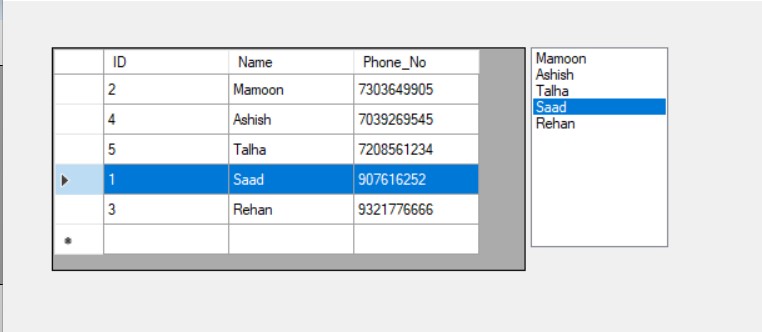
adpt.Fill(ds, "Visual") DataGridView1.DataSource = ds

DataGridView1.DataMember = "Visual"

ListBox1.DataSource = ds

ListBox1.DisplayMember = "Visual.name"

End Sub



# Practical 25 & 26

Develop a Windows application that will contain multiple table in a single dataset

Imports System.Data.OleDb

Public Class Form1

Dim cmd As OleDbCommand = New OleDbCommand

Dim reader As OleDbDataReader

Dim conn As OleDbConnection = New OleDbConnection

Dim adpt1 As OleDbDataAdapter = New OleDbDataAdapter

Dim adpt2 As OleDbDataAdapter = New OleDbDataAdapter

Dim ds As New DataSet

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load conn = New

OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Users\siddi\OneDrive\Documents\Database2.mdb;") conn.Open()

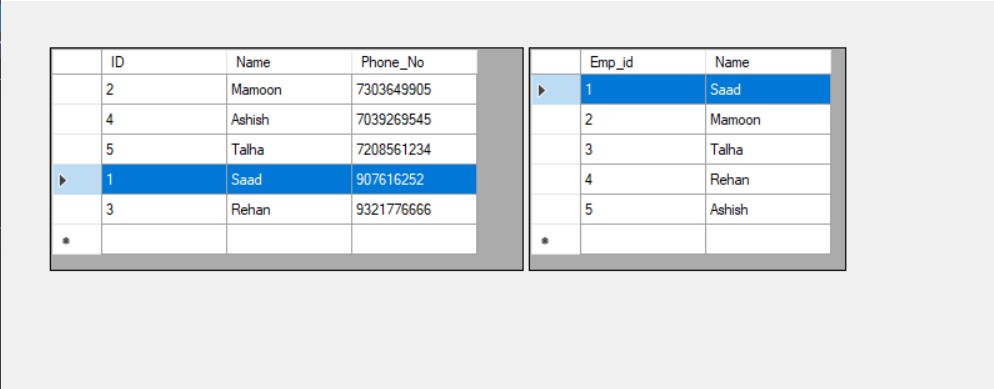
cmd = New OleDbCommand("select \* from Visual", conn) adpt1 = New OleDbDataAdapter(cmd)

adpt2 = New OleDbDataAdapter("select \* from Employee ", conn) adpt1.Fill(ds, "Visual") adpt2.Fill(ds, "Employee") DataGridView1.DataSource = ds

DataGridView1.DataMember = "Visual"

DataGridView2.DataSource = ds

DataGridView2.DataMember = "Employee" End Sub



Design a Windows application in MS-Access which have navigation (Next, First, Previous, Last)

Public Class Form1

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles

MyBase.Load

'TODO: This line of code loads data into the

'Database2DataSet.Visual\_S' table. You can move, or remove it, as needed.

Me.Visual\_STableAdapter.Fill(Me.Database2DataSet.Visual\_S)

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs)

Handles Button1.Click

VisualSBindingSource.MoveNext()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs)

Handles Button2.Click

VisualSBindingSource.MovePrevious()

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs)

Handles Button3.Click

VisualSBindingSource.MoveFirst()

End Sub

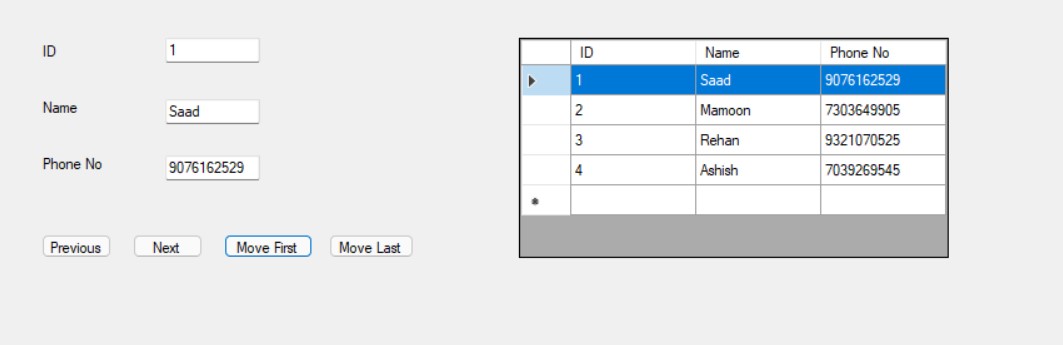
Private Sub Button4\_Click(sender As Object, e As EventArgs)

Handles Button4.Click

VisualSBindingSource.MoveLast()

End Sub

End Class



# Practical 27

Design a simple Windows Form for accepting the detail of Employee. Using the connected architecture of ADO.NET, perform the following operation

* Insert record
* Search record
* Update record
* Delete record

Imports System.Data.OleDb

Public Class Form1

Dim cmd As OleDbCommand = New OleDbCommand

Dim reader As OleDbDataReader

Dim conn As OleDbConnection = New OleDbConnection

Dim adpt As OleDbDataAdapter = New OleDbDataAdapter

Dim ds As New DataSet

Dim str As String

Dim MaxRow, inc As Integer

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles MyBase.Load conn = New

OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Users\siddi\OneDrive\Documents\Database2.mdb") conn.Open()

cmd = New OleDbCommand("select \* from Visual", conn) adpt = New OleDbDataAdapter(cmd) adpt.Fill(ds, "Visual") DataGridView1.DataSource = ds DataGridView1.DataMember = "Visual" reader = cmd.ExecuteReader()

reader.Read()

MaxRow = ds.Tables("Visual").Rows.Count inc = 1

TextBox1.Text = reader.GetValue(0)

TextBox2.Text = reader.GetValue(1) TextBox3.Text = reader.GetValue(2) conn.Close()

End Sub

Private Sub Label1\_Click(sender As Object, e As EventArgs) Handles

Label1.Click

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs)

Handles Button1.Click

TextBox1.Text = ""

TextBox2.Text = ""

TextBox3.Text = ""

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs)

Handles Button2.Click conn.Open() Dim count As Integer

str = "Insert into Visual values(" & CInt(TextBox1.Text) & ",

'" & TextBox2.Text & "'," & CDbl(TextBox3.Text) & ")" cmd = New OleDbCommand(str, conn) count = cmd.ExecuteNonQuery() MsgBox(count & " Record saved") conn.Close()

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs)

Handles Button3.Click conn.Open()

If MessageBox.Show("Do you really want to delete this record",

"Delete", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) =

Windows.Forms.DialogResult.No Then

MsgBox("Operation cancelled")

Exit Sub End If

str = "delete from Visual where ID = " & (TextBox1.Text) & ""

cmd = New OleDbCommand(str, conn)

cmd.ExecuteNonQuery() MsgBox("Record deleted")

conn.Close()

End Sub

Private Sub Button6\_Click(sender As Object, e As EventArgs)

Handles Button6.Click If inc > 0 Then inc = inc - 1

TextBox1.Text = ds.Tables("Visual").Rows(inc).Item(0) TextBox2.Text = ds.Tables("Visual").Rows(inc).Item(1)

TextBox3.Text = ds.Tables("Visual").Rows(inc).Item(2)

Else

MessageBox.Show("First Record")

End If

End Sub

Private Sub Button5\_Click(sender As Object, e As EventArgs)

Handles Button5.Click

If inc <> MaxRow Then

TextBox1.Text = ds.Tables("Visual").Rows(inc).Item(0)

TextBox2.Text = ds.Tables("Visual").Rows(inc).Item(1) TextBox3.Text = ds.Tables("Visual").Rows(inc).Item(2) inc = inc + 1

Else

MessageBox.Show("No more rows")

End If

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs)

Handles Button4.Click conn.Open()

str = "update Visual set Name = " & (TextBox2.Text) & ", Phone\_No = " & CDbl(TextBox3.Text) & ", ID = " & CInt(TextBox1.Text) &

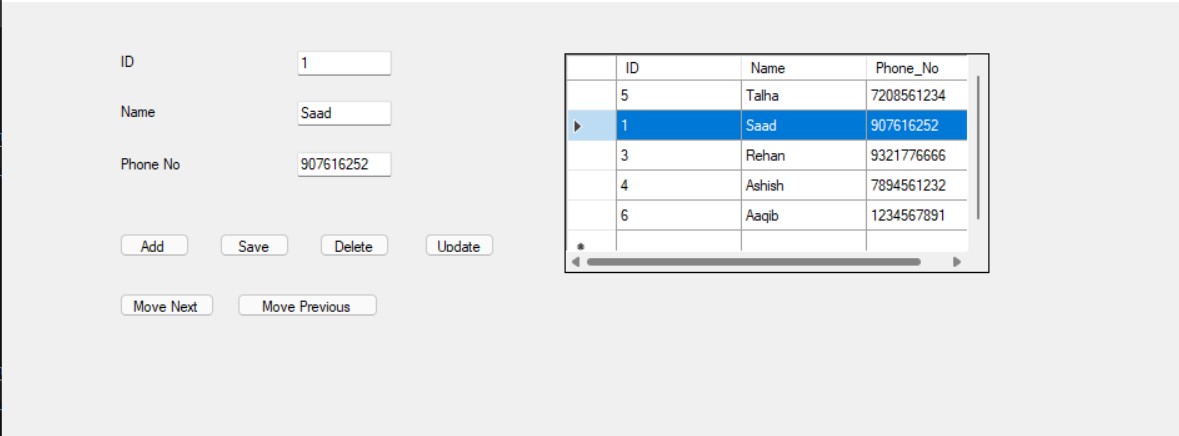
""

cmd = New OleDbCommand(str, conn)

conn.Close()

End Sub

End Class



# Practical 13 & 14

Write a Program to perform validation using ErrorProvider & Regular Expression **Or**

Write a Program code to perform Date Validation using ErrorProvider Control **Or**

Write a Program using ErrorProvider control to Validate the Mobile

Number and Email ID in GUI application

Public Class Form1

Private Sub TextBox1\_validating(sender As Object, e As EventArgs)

Handles TextBox1.TextChanged

Dim check As New System.Text.RegularExpressions.Regex("[A-Zaz]")

If check.IsMatch(TextBox1.Text) Then

ErrorProvider1.SetError(TextBox1, "")

Else

ErrorProvider1.SetError(TextBox1, "Not a valid name")

End If

End Sub

Private Sub TextBox2\_validating(sender As Object, e As EventArgs)

Handles TextBox2.TextChanged

Dim check As New System.Text.RegularExpressions.Regex("(0[1-

9]|1[0-9]|2[0-9]|3[01])[-/.](0[1-9]|1[012])[-/.][19|20]\d\d")

If check.IsMatch(TextBox2.Text) Then

ErrorProvider1.SetError(TextBox2, "")

Else

ErrorProvider1.SetError(TextBox2, "Not a valid date")

End If

End Sub

Private Sub TextBox3\_validating(sender As Object, e As EventArgs)

Handles TextBox3.TextChanged

Dim check As New

System.Text.RegularExpressions.Regex("\S+@\S+.\S")

If check.IsMatch(TextBox3.Text) Then

ErrorProvider1.SetError(TextBox3, "")

Else

ErrorProvider1.SetError(TextBox3, "Not a Valid Mail") End If

End Sub

Private Sub TextBox4\_validating(sender As Object, e As EventArgs)

Handles TextBox4.TextChanged

Dim check As New System.Text.RegularExpressions.Regex("[09]\*$")

If check.IsMatch(TextBox4.Text) And TextBox4.TextLength = 10

Then

ErrorProvider1.SetError(TextBox4, "")

Else

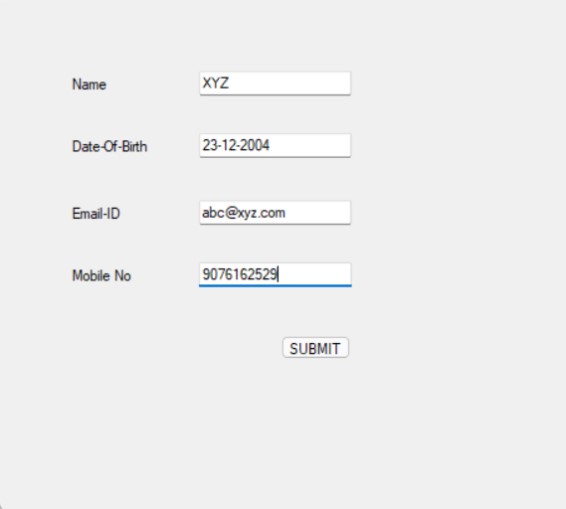
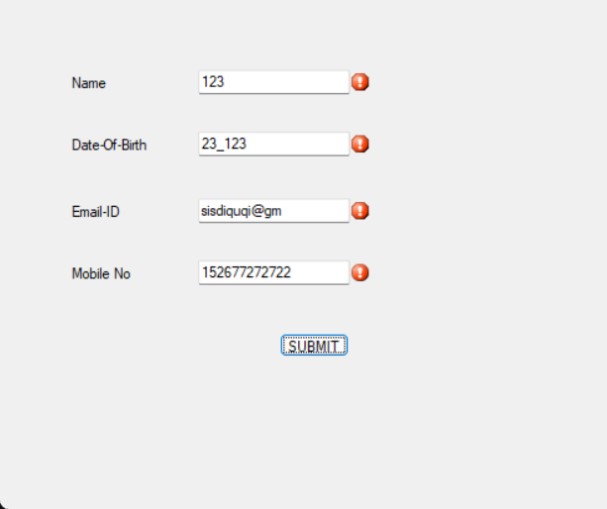
ErrorProvider1.SetError(TextBox4, "Not a Valid Mobile

Number")

End If

End Sub

End Class



Write a Program using ErrorProvider for username & password authentication

Public Class Form4

Private Sub TextBox1\_validating(sender As Object, e As EventArgs)

Handles TextBox1.TextChanged

Dim check As New System.Text.RegularExpressions.Regex("([A-

Z]|[a-z])(|[\_.])(\d)")

If check.IsMatch(TextBox1.Text) Then

ErrorProvider1.SetError(TextBox1, "")

Else

ErrorProvider1.SetError(TextBox1, "Not a Valid Username")

End If

End Sub

Private Sub TextBox2\_validating(sender As Object, e As EventArgs) Handles TextBox2.TextChanged

Dim check As New System.Text.RegularExpressions.Regex("([A-

Z]|[a-z])([0-9])")

If check.IsMatch(TextBox2.Text) Then

ErrorProvider1.SetError(TextBox2, "")

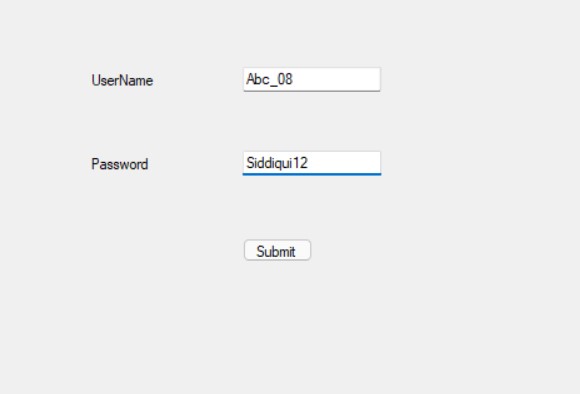
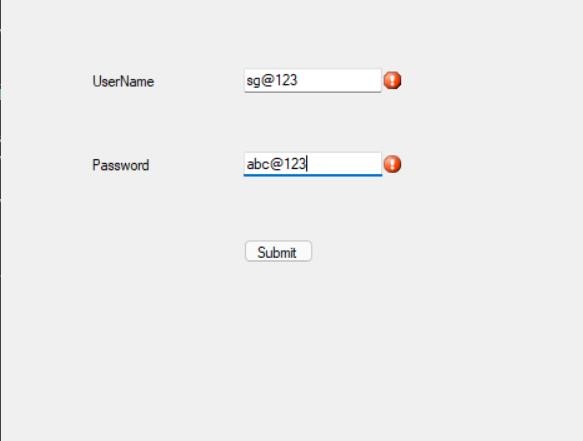
Else

ErrorProvider1.SetError(TextBox2, "Not a Valid Password")

End If

End Sub

End Class



# Practical 27 , 28 , 29

Q1) Design a window application for student name and college name using a simple data binding use appropriate database

Imports System.Data.OleDb

Public Class Form5

Dim cmd As OleDbCommand = New OleDbCommand

Dim conn As OleDbConnection = New

OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data

Source=C:\Users\Talha \Documents\GAD.mdb")

Dim ds As New DataSet

Private Sub Form5\_Load(sender As Object, e As EventArgs) Handles MyBase.Load conn.Open()

cmd = New OleDbCommand("select \* from College where

STUDENT\_NAME='Matt' ", conn)

Dim adpt As New OleDbDataAdapter(cmd)

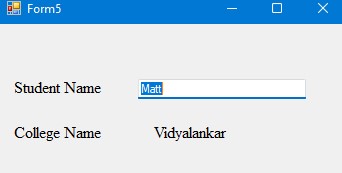
adpt.Fill(ds, "College")

TextBox1.DataBindings.Add("text", ds, "College.STUDENT\_NAME")

Label2.DataBindings.Add("text", ds, "College.COLLEGE\_NAME")

End Sub

End Class



Q2. Design a window application for bank customer record & display it using Complex data binding use appropriate database

Imports System.Data.OleDb

Public Class Form6

Dim conn As OleDbConnection = New

OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data

Source=C:\Users\Talha\Documents\GAD.mdb") ' Connection

Dim cmd As OleDbCommand = New OleDbCommand

Dim reader As OleDbDataReader

Dim adpt As OleDbDataAdapter = New OleDbDataAdapter

Dim ds As New DataSet

Private Sub Form6\_Load(sender As Object, e As EventArgs) Handles MyBase.Load conn.Open()

cmd = New OleDbCommand("select \* from Bank", conn) adpt = New OleDbDataAdapter(cmd)

adpt.Fill(ds, "Bank")

DataGridView1.DataSource = ds DataGridView1.DataMember =

"Bank"ComboBox1.DataSource = ds

ComboBox1.DisplayMember = "Bank.ACCOUNT\_NO"

ComboBox2.DataSource = ds ComboBox2.DisplayMember =

"Bank.CUSTOMER\_NAME"

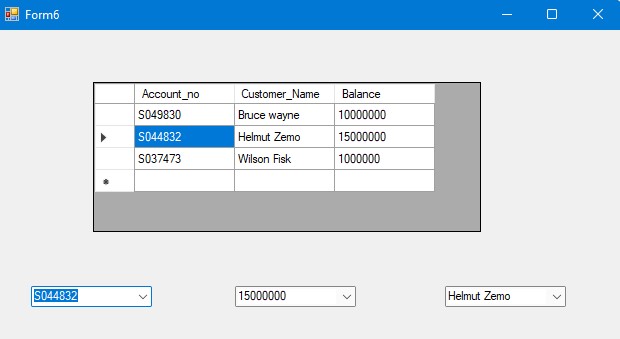
ComboBox3.DataSource = ds

ComboBox3.DisplayMember = "Bank.BALANCE"

conn.Close()

End Sub

End Class



# Practical 31

Q1. Design the window application using MS-Access database table with name student & navigate to First, Next, Previous, Last records

Imports System.Data.OleDb

Public Class Form7

Dim cmd As OleDbCommand = New OleDbCommand

Dim reader As OleDbDataReader

Dim conn As OleDbConnection = New OleDbConnection

Dim adpt As OleDbDataAdapter = New OleDbDataAdapter

Dim ds As New DataSet

Dim inc, MaxRows As Integer

Private Sub Form7\_Load(sender As Object, e As EventArgs) Handles MyBase.Load conn = New

OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data

Source=C:\Users\Talha\Documents\GAD.mdb") conn.Open()

cmd = New OleDbCommand("select \* from Student", conn) adpt = New OleDbDataAdapter(cmd) adpt.Fill(ds, "Student") DataGridView1.DataSource = ds DataGridView1.DataMember = "Student" reader = cmd.ExecuteReader() reader.Read()

TextBox1.Text = reader.GetValue(0)

TextBox2.Text = reader.GetValue(1)

TextBox3.Text = reader.GetValue(2) MaxRows = ds.Tables("Student").Rows.Count inc = 1 conn.Close()

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs)

Handles Button2.Click

If inc <> MaxRows Then

TextBox1.Text = ds.Tables("Student").Rows(inc).Item(0)

TextBox2.Text = ds.Tables("Student").Rows(inc).Item(1) TextBox3.Text = ds.Tables("Student").Rows(inc).Item(2) inc = inc + 1

Else

MessageBox.Show("You have reached to last record ") End If

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs)

Handles Button4.Click If inc > 0 Then inc = inc - 1

TextBox1.Text = ds.Tables("Student").Rows(inc).Item(0)

TextBox2.Text = ds.Tables("Student").Rows(inc).Item(1)

TextBox3.Text = ds.Tables("Student").Rows(inc).Item(2) Else

MessageBox.Show("You have reached to First Record")

End If

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs)

Handles Button1.Click If inc <> 0 Then inc = 0

TextBox1.Text = ds.Tables("Student").Rows(inc).Item(0)

TextBox2.Text = ds.Tables("Student").Rows(inc).Item(1)

TextBox3.Text = ds.Tables("Student").Rows(inc).Item(2)

End If

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs)

Handles Button3.Click

If inc <> MaxRows - 1 Then inc = MaxRows - 1

TextBox1.Text = ds.Tables("Student").Rows(inc).Item(0)

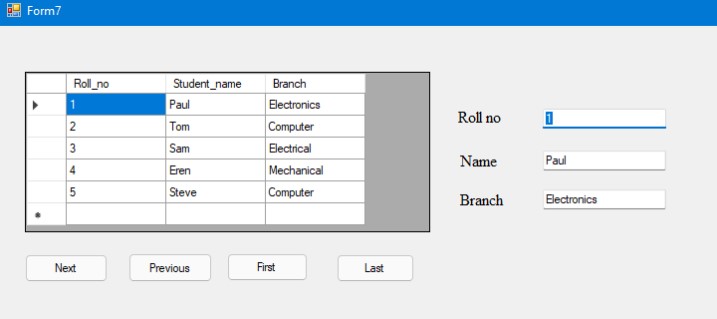
TextBox2.Text = ds.Tables("Student").Rows(inc).Item(1)

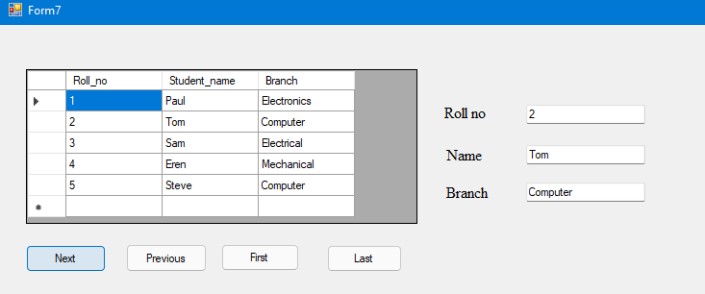
TextBox3.Text = ds.Tables("Student").Rows(inc).Item(2)

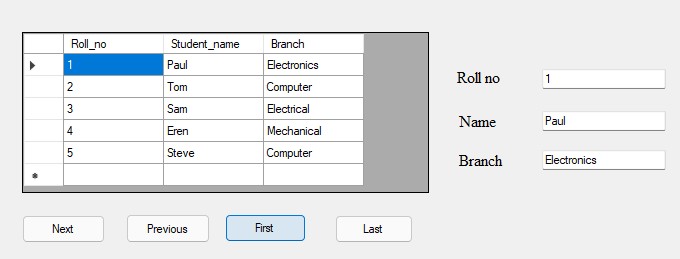
End If

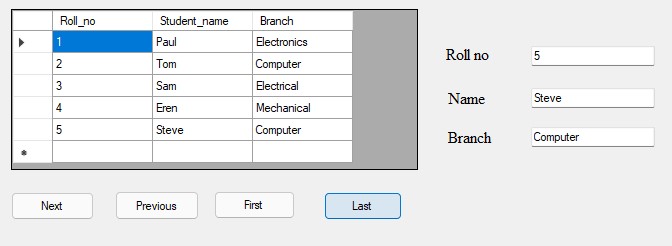
End Sub

End Class









Q) Write a Program to Navigate in the Database

Imports System.Data.OleDb

Public Class Form2

Dim cmd As New OleDbCommand

Dim reader As OleDbDataReader

Dim conn As New OleDbConnection

Dim adpt As New OleDbDataAdapter

Dim ds As New DataSet

Dim maxrow, inc As Integer

Private Sub Form2\_Load(sender As Object, e As EventArgs) Handles MyBase.Load conn = New

OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\Users\siddi\OneDrive\Documents\Database2.mdb") conn.Open()

cmd = New OleDbCommand("select \* from navigate", conn) adpt = New OleDbDataAdapter(cmd) adpt.Fill(ds, "navigate") reader = cmd.ExecuteReader() reader.Read()

TextBox1.Text = reader.GetValue(0)

TextBox2.Text = reader.GetValue(1) TextBox3.Text = reader.GetValue(2) maxrow = ds.Tables("navigate").Rows.Count inc = 1

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs)

Handles Button1.Click If inc <> 0 Then inc = 0

TextBox1.Text = ds.Tables("navigate").Rows(inc).Item(0)

TextBox2.Text = ds.Tables("navigate").Rows(inc).Item(1) TextBox3.Text = ds.Tables("navigate").Rows(inc).Item(2)

End If

End Sub

Private Sub Button2\_Click(sender As Object, e As EventArgs)

Handles Button2.Click If inc <> maxrow Then inc = inc + 1

TextBox1.Text = ds.Tables("navigate").Rows(inc).Item(0) TextBox2.Text = ds.Tables("navigate").Rows(inc).Item(1) TextBox3.Text = ds.Tables("navigate").Rows(inc).Item(2)

End If

End Sub

Private Sub Button3\_Click(sender As Object, e As EventArgs)

Handles Button3.Click If inc > 0 Then inc = inc - 1

TextBox1.Text = ds.Tables("navigate").Rows(inc).Item(0)

TextBox2.Text = ds.Tables("navigate").Rows(inc).Item(1)

TextBox3.Text = ds.Tables("navigate").Rows(inc).Item(2)

Else

MessageBox.Show("First Record")

End If

End Sub

Private Sub Button4\_Click(sender As Object, e As EventArgs)

Handles Button4.Click

If inc <> maxrow - 1 Then inc = maxrow - 1

TextBox1.Text = ds.Tables("navigate").Rows(inc).Item(0)

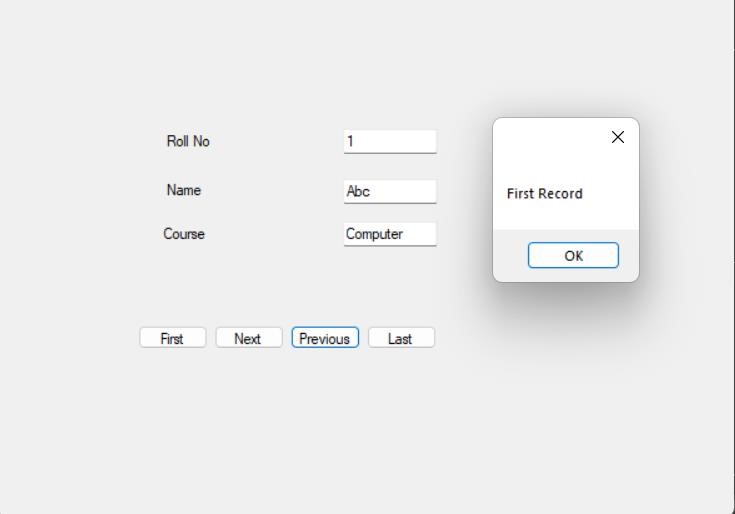
TextBox2.Text = ds.Tables("navigate").Rows(inc).Item(1)

TextBox3.Text = ds.Tables("navigate").Rows(inc).Item(2)

End If

End Sub

End Class



# Practical No 15

Implement a windows application using sub-procedures & parameterized sub-procedures.

Write a Program to calculate the reverse of a given number

Code:

Module Module1

Sub Main()

Dim n As Integer

Console.Write("enter number") n = CInt(Console.ReadLine()) reverse(n) Console.ReadLine()

End Sub

Sub reverse(ByVal n As Integer)

Dim i, rev As Integer

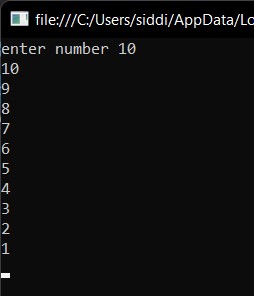
For i = n To 1 Step -1

Console.WriteLine(i)

Next

End Sub

End Module Output:



Write a Program to calculate the Fibonacci series of a given number

Code:

Module Module1

Sub Main()

Dim n As Integer

Console.Write("Enter how many elements:-")

n = Val(Console.ReadLine()) fibonicci(n) Console.WriteLine()

End Sub

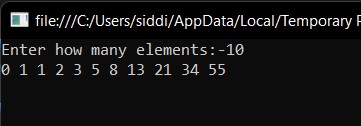
Sub fibonicci(ByVal n As Integer) Dim i, a, b, c As Integer a = 0 b = 1 Console.Write("0 1") i = 0 While (i < n - 1) c = a + b Console.Write(" " & c) a = b b = c i = i + 1

End While

Console.ReadLine()

End Sub

End Module Output:



# Practical No 16

Implement a program to demonstrate use of simple functions & parameterized sub-functions.

Code:

Public Class Form1

Function Greater(x As Integer, y As Integer) As Integer

Dim result As Integer If x > y Then result = x

TextBox3.Text = result Else result = y

TextBox3.Text = result

End If

Return result

End Function

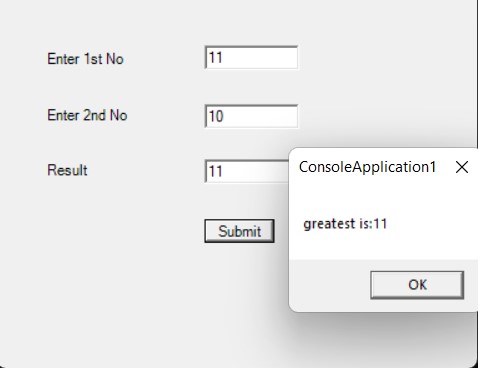
Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles

Button1.Click

Dim x, y, result As Integer x = TextBox1.Text y = TextBox2.Text result = Greater(x, y) MsgBox("greatest is:" & result)

End Sub

End Class Output:

 Code:

Public Class Form2

Dim fact As Integer = 0

Dim num, f As Integer

Function Factorial(ByVal num As Integer)

If num = 1 Then

Return 1 Else

f = Factorial(num - 1) \* num

Return f

End If

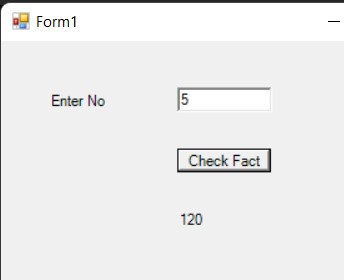
End Function

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click num = TextBox1.Text f = Factorial(num)

Label3.Text = f

End Sub

End Class Output:



# Practical No 17

Understand the Concept of Class and Object of Class.

Code:

Module Module1

Sub main() Dim v As New box

v.volume(5, 5, 5) Console.ReadLine()

End Sub

Class box

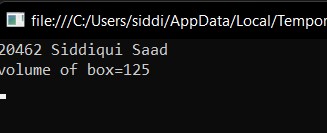
Sub volume(ByVal l As Integer, ByVal b As Integer, ByVal h As Integer)

Dim vol As Integer vol = l \* b \* h

Console.WriteLine("volume of box=" & vol)

End Sub

End Class End Module Output:



Code:

Public Class Form1

Private Sub Form1\_Load(sender As Object, e As EventArgs) Handles

MyBase.Load

ComboBox1.Items.Add("2")

ComboBox1.Items.Add("3")

ComboBox1.Items.Add("4")

ComboBox2.Items.Add("5")

ComboBox2.Items.Add("6")

ComboBox2.Items.Add("7")

ComboBox3.Items.Add("8")

ComboBox3.Items.Add("9")

ComboBox3.Items.Add("10")

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles

Button1.Click

Dim N As Integer()

N = {ComboBox1.SelectedItem, ComboBox2.SelectedItem,

ComboBox3.SelectedItem}

Dim OBJ As New AVERAGE

OBJ.AVG(N)

End Sub

End Class

Class AVERAGE

Function AVG(ByVal ParamArray N As Integer())

Dim S, I, A As Integer

For Each I In N

S = S + I

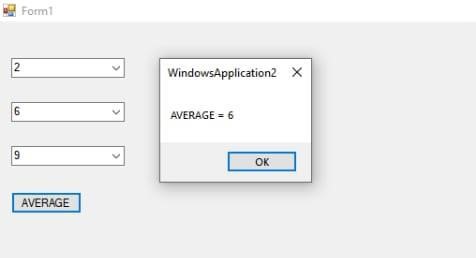
Next

A = S / 3

MsgBox("AVERAGE = " & A)

Return 0

End Function End Class Output:



# Practical No: 18

Implement a program for class constructor and destructor to de-allocate memory.

Code:

Module Module1

Sub Main()

Dim s As New display()

Console.ReadLine()

End Sub

Class display

Public Sub New()

Console.WriteLine(" Display msg: Constructor called")

End Sub

End Class End Module Output:

 Code:

Module Module1

Sub Main()

Dim c As New circle(5)

Console.ReadLine()

End Sub

Class circle

Dim pi As Double = 3.14

Dim a As Double

Sub New(ByVal r As Integer) a = 2 \* pi \* r

Console.WriteLine("Area of a circle:" & a)

End Sub

End Class End Module Output:



# Practical No: 19

Develop a program for Inheritance

Code:

Module Module3

Public Class student

Public name1 As String

Public rollno As Integer

Public Sub New(ByVal a As String, ByVal b As Integer) name1 = a rollno = b End Sub

End Class

Public Class marks : Inherits student

Public sub1 As Integer

Public Sub2 As Integer

Public Sub New(ByVal a As String, ByVal b As Integer, ByVal c As

Integer, ByVal d As Integer)

MyBase.New(a, b) 'passing value to base class student constructor sub1 = c Sub2 = d

End Sub

Function average() As Double Dim s As Double s = (sub1 + Sub2) / 2

Return s

End Function

End Class

Public Class result : Inherits marks

Public division As String

Public Sub New(ByVal a As String, ByVal b As Integer, ByVal c As

Integer, ByVal d As Integer)

MyBase.New(a, b, c, d) 'passing value to base class marks constructor

End Sub

Function div() As String Dim p As Double p = average() If (p >= 60) Then division = "first" Else

division = "second" End If

Return division

End Function

End Class

Sub main()

Dim obj As New result("Sheeba", 59, 70, 80)

Console.WriteLine("name of student:" & obj.name1)

Console.WriteLine("roll n of student " & obj.rollno)

Console.WriteLine("subject1 of student " & obj.sub1)

Console.WriteLine("subject2 of student " & obj.Sub2)

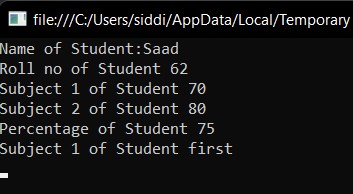
Console.WriteLine("percentage of student " & obj.average)

Console.WriteLine("subject1 of student " & obj.div)

Console.ReadLine()

End Sub

End Module Output:



# Practical No: 20 & 21

Implement a program for overloading and overriding

Code:

Module Module3

Class overloading

Sub volume(ByVal s As Integer)

Dim v As Integer v = s \* s \* s

Console.WriteLine("volume of box= " & v)

End Sub

Sub volume(ByVal l As Integer, ByVal b As Integer)

Dim v As Integer v = 2 \* (l \* b)

Console.WriteLine("volume of rectangle= " & v)

End Sub

Sub volume(ByVal r As Double, ByVal h As Integer) Dim v As Integer v = 3.14 \* r \* r \* r

Console.WriteLine("volume of cylinder= " & v)

End Sub

End Class

Class abc

Overridable Sub show()

Console.WriteLine("Inside super class")

End Sub

End Class

Class overriding : Inherits abc

Public Overrides Sub show()

MyBase.show()

Console.WriteLine("This statement is printed by overriding Volume of

Different shapes are printed through overloading")

End Sub

Private Sub are(p1 As Object)

Throw New NotImplementedException

End Sub

Private Function printed() As Object

Throw New NotImplementedException

End Function

End Class

Sub main()

Dim o As New overloading

Dim ob As New overriding

ob.show()

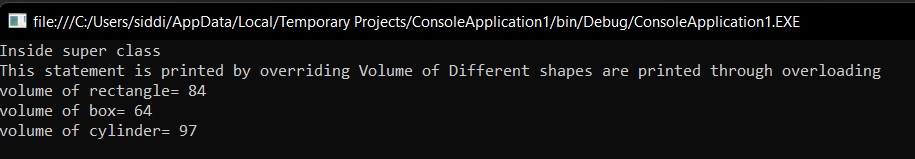
o.volume(6, 7)

o.volume(4)

o.volume(3.14, 10) Console.ReadLine()

End Sub

End Module Output:



Code:

Module Module3

Class overloading

Public n As String

Sub display(ByVal s As String)

s = n + s

Console.WriteLine(s)

End Sub

Sub display()

Console.WriteLine("Enter first name: ")

n = Console.ReadLine()

End Sub

End Class

Sub main()

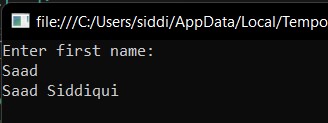
Dim o As New overloading o.display()

o.display(" Siddiqui ")

Console.ReadLine()

End Sub

End Module Output:



# Practical No: 22

Implement a program to demonstrate Shadowing in inheritance.

Code:

Module Module1

Class A

Public Sub show()

Console.WriteLine("Calling from A")

End Sub

End Class

Class B: Inherits A

Private Shadows Sub show()

Console.WriteLine("Calling from B")

End Sub

End Class

Class C: Inherits B

Private Shadows Sub show()

Console.WriteLine("Calling from C")

End Sub

End Class

Sub Main()

Dim x As New A

Dim y As New B Dim z As New C

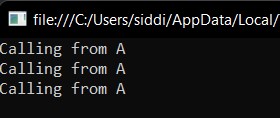
x.show()

y.show()

z.show() Console.ReadLine()

End Sub

End Module Output:



# Practical No: 23

Implement a program to handle runtime errors using Exception handling

Code:

Module Module3

Sub division(ByVal num1 As Integer, ByVal num2 As Integer)

Dim result As Integer Try

result = num1 \ num2

Catch e As DivideByZeroException

Console.WriteLine("Exception caught: {0}", e)

Finally

Console.WriteLine("Result: {0}", result)

End Try

End Sub

Sub Main()

Dim num1, num2 As Integer num1 = InputBox("Enter Number 1") num2 = InputBox("Enter Number 2")

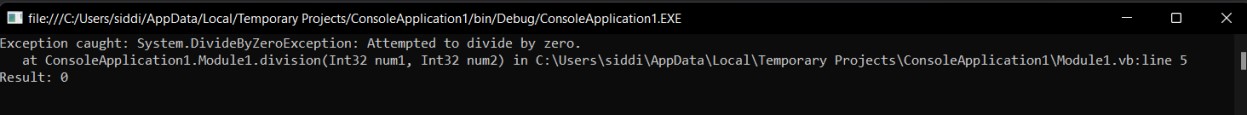
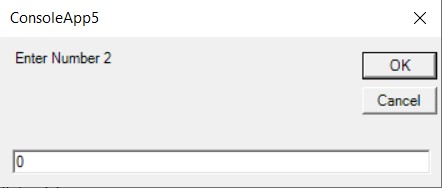
division(num1, num2)

Console.ReadLine()

End Sub

End Module Output:

Code:



Module Module3

Sub Main()

Dim a, b, c As Integer

Console.WriteLine("Enter 2 numbers") a = Console.ReadLine() b = Console.ReadLine() Try c = a / b

Catch ex As Exception

Console.WriteLine("DO NOT ENTER 0")

Console.ReadLine()

Return

End Try

Console.WriteLine("ANSWER = " & c)

Console.ReadLine()

End Sub

End Module Output:

