****

**Object Oriented Analysis and Design (SS ZG514)**

**Assignment – Phase 2**

Submitted By:

Nilesh D. Ghodekar (2018ht12544)

Set2\_Group\_New6

Q 1. Identify at least **two GOF Design Patterns** that can be used. Provide details of how these patterns can be applied in order to improve the Object-Oriented design of the system. Also provide implementation (Java code or pseudo code) of at least one of them.

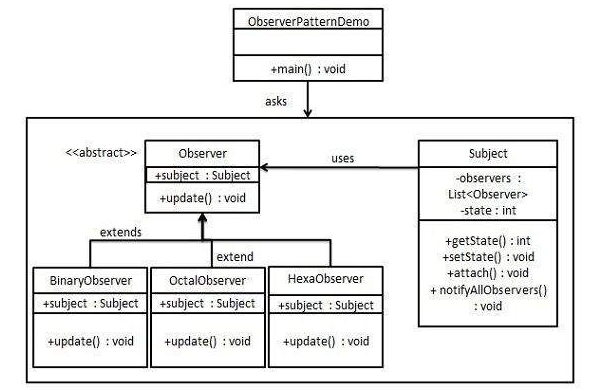
**I. Observer pattern**

It is used when there is one to many relationship between objects such as if one object is modified, its dependent objects are to be notified automatically. Observer pattern falls under behavioral pattern category.

## Implementation

Observer pattern uses three actor classes. Subject, Observer and Client. Subject, an object having methods to attach and de-attach observers to a client object. We've created classes Subject, Observer abstract class and concrete classes extending the abstract class the Observer.

Design Diagram:



**Implementation Code:**

Public Class MdiForm

Private Sub EmployeeToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles EmployeeToolStripMenuItem1.Click

Dim emp As New Employee

emp.Show()

emp.MdiParent = Me

End Sub

Private Sub CustomerToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CustomerToolStripMenuItem1.Click

Dim cust As New Customer

cust.Show()

cust.MdiParent = Me

End Sub

Private Sub SupplierToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SupplierToolStripMenuItem1.Click

Dim sup As New Supplier

sup.Show()

sup.MdiParent = Me

End Sub

Private Sub PurchaseOrderToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PurchaseOrderToolStripMenuItem1.Click

Dim pur As New PurchaseOrder

pur.Show()

pur.MdiParent = Me

End Sub

Private Sub SalesOrderToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SalesOrderToolStripMenuItem1.Click

Dim sales As New SalesOrder

sales.Show()

sales.MdiParent = Me

End Sub

Private Sub CylinderDeliveryToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CylinderDeliveryToolStripMenuItem1.Click

Dim cdel As New CylinderDelivery

cdel.Show()

cdel.MdiParent = Me

End Sub

Private Sub SalesOrderReportToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SalesOrderReportToolStripMenuItem1.Click

Dim salesrpt As New SOrder

salesrpt.Show()

salesrpt.MdiParent = Me

nd Sub

Private Sub CylinderDeliveryReportToolStripMenuItem\_Click

(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CylinderDeliveryReportToolStripMenuItem.Click

Dim cdelrpt As New CylDel

cdelrpt.Show()

cdelrpt.MdiParent = Me

End Sub

Private Sub ChangePasswordToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ChangePasswordToolStripMenuItem1.Click

Dim changepasswrd As New ChangePassword

changepasswrd.Show()

changepasswrd.MdiParent = Me

End Sub

Private Sub AboutUsToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutUsToolStripMenuItem1.Click

Dim abt As New AboutUs

abt.Show()

abt.MdiParent = Me

End Sub

Private Sub LogOutToolStripMenuItem1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles LogOutToolStripMenuItem1.Click

Dim ans As Integer

ans = MsgBox("Do You Really Want To Exit?", MsgBoxStyle.Question + MsgBoxStyle.YesNo, "EXIT")

If ans = MsgBoxResult.No Then

Exit Sub

Else

End

End If

End Sub

End Class

**Employee.vb**

Imports System.Data.SqlClient

Public Class Employee

Dim objCon As New SqlConnection("Data Source=.;Initial Catalog=gasagency;Integrated Security=True")

Dim objDataAdapter As New SqlDataAdapter("select \* from Employee",

objCon)

Dim objDataSet As DataSet

Dim objDataView As DataView

Dim objCurrencyManager As CurrencyManager

Private Sub fillDataSetAndView()

objDataSet = New DataSet()

objDataAdapter.Fill(objDataSet, "Employee")

objDataView = New DataView(objDataSet.Tables("Employee"))

objCurrencyManager = CType(Me.BindingContext(objDataView), CurrencyManager)

End Sub

Private Sub bindFields()

txteid.DataBindings.Clear()

txtenm.DataBindings.Clear()

txtphno.DataBindings.Clear()

rchad.DataBindings.Clear()

txtsal.DataBindings.Clear()

txteid.DataBindings.Add("Text", objDataView, "EmpId")

txtenm.DataBindings.Add("Text", objDataView, "EmpNm")

txtphno.DataBindings.Add("Text", objDataView, "EmpContact")

rchad.DataBindings.Add("Text", objDataView, "EmpAdd")

txtsal.DataBindings.Add("Text", objDataView, "EmpSal")

End Sub

'--------------Following code for-------------

Grid View

'-------------------------------------------------

Public Sub viewrecord()

objCon = New SqlConnection("Data Source=.;Initial

Catalog=gasagency;

Integrated Security=True")

objCon.Open()

objDataAdapter = New SqlDataAdapter("select\* from Employee", objCon)

objDataSet = New DataSet

objDataAdapter.Fill(objDataSet, "Employee")

DataGridView1.DataSource = objDataSet

DataGridView1.DataMember = "Employee"

objCon.Close()

End Sub

Private Sub showPosition()

txtRecord.Text= objCurrencyManager.Position + 1 & " of " &

objCurrencyManager.Count()

End Sub

Private Sub Employee\_Load (ByVal sender As System.EventArgs)

Handles MyBase.Load

fillDataSetAndView()

bindFields()

showPosition()

viewrecord()

txteid.Enabled = False

txtenm.Enabled = False

rchad.Enabled = False

txtphno.Enabled = False

txtsal.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

DataGridView1.ReadOnly = True

DataGridView1.ReadOnly = True

End Sub

Private Sub btnfrst\_Click(ByVal sender As System.Object, ByVal e AsSystem.EventArgs)Handles btnfrst.Click

fillDataSetAndView()

bindFields()

objCurrencyManager.Position = 0

showPosition()

txteid.Enabled = False

txtenm.Enabled = False

txtphno.Enabled = False

rchad.Enabled = False

txtsal.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnnxt\_Click(ByVal sender As System.Object, ByVal e AsSystem.EventArgs) Handles btnnxt.Click

objCurrencyManager.Position += 1

showPosition()

txteid.Enabled = False

txtenm.Enabled = False

txtphno.Enabled = False

rchad.Enabled = False

txtsal.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnprev\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnprev.Click

objCurrencyManager.Position -= 1

showPosition()

txteid.Enabled = False

txtenm.Enabled = False

txtphno.Enabled = False

rchad.Enabled = False

txtsal.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnlast\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnlast.Click

objCurrencyManager.Position = objCurrencyManager.Count() - 1

showPosition()

txteid.Enabled = False

txtenm.Enabled = False

txtphno.Enabled = False

rchad.Enabled = False

txtsal.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnsrch\_Click(ByVal sender As System.Object, ByVal e AsSystem.EventArgs) Handles btnsrch.Click

Dim i As Integer

objDataView.Sort = "EmpId"

i = objDataView.Find(txtsrch.Text)

showPosition()

If (i = -1) Then

MessageBox.Show("SORRY!!Record not found!!!")

txtsrch.Text = ""

Else

objCurrencyManager.Position = i

txtsrch.Text = ""

End If

showPosition()

End Sub

Private Sub btnnew\_Click(ByVal sender As System.Object, ByVal e AsSystem.EventArgs) Handles btnnew.Click

bindFields()

txteid.Text = ""

'----------------Following code for--------------

Automatically Generate Employee ID

'--------------------------------------------------

Dim strId As String

Dim MaxID As String = GetMaxID()

If MaxID = "" Then

txteid.Text = "E101"

Else

strId = CType(MaxID.Remove(0, 1), String)

MaxID = CType(strId, Integer)

MaxID += 1

strId = "E" & MaxID.ToString

txteid.Text = strId

End If

txteid.Enabled = False

txtenm.Text = ""

txtphno.Text = ""

rchad.Text = "" txtsal.Text = ""

txtenm.Enabled = True

txtphno.Enabled = True

rchad.Enabled = True

txtsal.Enabled = True

btnadd.Enabled = True

End Sub

Public Function GetMaxID() As String

Dim ds As DataSet = New DataSet

Dim objCon As New SqlConnection("Data Source=.;Initial Catalog=gasagency;Integrated Security=True")

Dim cmd As SqlCommand = New SqlCommand("select MAX(EmpId)from

Employee", objCon)

objCon.Open()

Dim i As String = cmd.ExecuteScalar().ToString()

Return i

End Function

Private Sub btnadd\_Click(ByVal sender As System.Object, ByVal e As

System.EventArgs) Handles btnadd.Click

Dim objCommand As SqlCommand = New SqlCommand

Dim i As Integer

i = objCurrencyManager.Position

If txtenm.Text.Trim() = String.Empty Then

MessageBox.Show("Please enter Employee Name!!!!!")

ElseIf txtphno.Text.Trim() = String.Empty Then

MessageBox.Show("Please enter Employee Phone Number!!!!!")

ElseIf rchad.Text.Trim() = String.Empty Then

MessageBox.Show("Please enter Employee Address!!!!!")

Else

objCon.Open()

objCommand.Connection = objCon

objCommand.CommandText="insert into Employee(EmpId,EmpNm,

EmpSal,EmpAdd ,EmpContact)Values(@EmpId,@EmpNm,@EmpSal,

@EmpAdd,@EmpContact)"

MessageBox.Show("The record has been added Sccessfully!!!")

objCommand.Parameters.AddWithValue("@EmpId", txteid.Text)

objCommand.Parameters.AddWithValue("@EmpNm", txtenm.Text)

objCommand.Parameters.AddWithValue("@EmpContact",txtphno

.Text)

objCommand.Parameters.AddWithValue("@EmpAdd", rchad.Text)

objCommand.Parameters.AddWithValue("@EmpSal", txtsal.Text)

Try

objCommand.ExecuteNonQuery()

Catch sqlExceptionErr As SqlException

MessageBox.Show(sqlExceptionErr.Message)

End Try

objCon.Close()

fillDataSetAndView()

bindFields()

objCurrencyManager.Position = i

showPosition()

viewrecord()

txtenm.Enabled = False

txtphno.Enabled = False

rchad.Enabled = False

txtsal.Enabled = False

btndelete.Enabled = False

btnupdate.Enabled = False

End If

End Sub

Private Sub btnupdate\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnupdate.Click

Dim objCommand As SqlCommand = New SqlCommand

objCon.Open()

objCommand.Connection = objCon

Dim i As Integer

i = objCurrencyManager.Position

objCommand.CommandText = "update Employee set EmpSal=@EmpSal,

EmpAdd=@EmpAdd,EmpContact=@EmpContact where EmpId=@EmpId"

MessageBox.Show("The record has been updated

successfully!!!")

objCommand.Parameters.AddWithValue("@EmpId", txteid.Text)

objCommand.Parameters.AddWithValue("@EmpNm", txtenm.Text)

objCommand.Parameters.AddWithValue("@EmpContact", txtphno.Text)

objCommand.Parameters.AddWithValue("@EmpAdd", rchad.Text)

objCommand.Parameters.AddWithValue("@EmpSal", txtsal.Text)

Try

objCommand.ExecuteNonQuery()

Catch sqlExceptionErr As SqlException

MessageBox.Show(sqlExceptionErr.Message)

End Try

objCon.Close()

fillDataSetAndView()

bindFields()

objCurrencyManager.Position = i

showPosition()

viewrecord()

txtenm.Enabled = False

txtphno.Enabled = False

rchad.Enabled = False

txtsal.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

End Sub

Private Sub btndelete\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btndelete.Click

Dim intposition As Integer

Dim objCommamnd As SqlCommand = New SqlCommand()

intposition = Me.BindingContext(objDataView).Position - 1

If intposition < 0 Then

intposition = 0

End If

Dim x = MsgBox("Are You Sure you want to delete this record", MsgBoxStyle.Question + MsgBoxStyle.YesNo)

If x = vbYes Then

objCommamnd.Connection = objCon

objCommamnd.CommandText = "delete from Employee where EmpId=@EmpId"

objCommamnd.Parameters.AddWithValue("@EmpId",BindingContext(objDataVie w).Current("EmpId"))

MessageBox.Show("The record has been deleted Successfully!!!!")

objCon.Open()

objCommamnd.ExecuteNonQuery()

txteid.Text = ""

txtenm.Text = ""

txtphno.Text = ""

rchad.Text = ""

txtsal.Text = ""

objCon.Close()

fillDataSetAndView()

bindFields()

viewrecord()

Me.BindingContext(objDataView).Position = intposition

showPosition()

txtenm.Enabled = False

txtphno.Enabled = False

rchad.Enabled = False

txtsal.Enabled = False

btnupdate.Enabled = False

btnadd.Enabled = False

EndIf

End Sub

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnexit.Click

Me.Close()

End Sub

Private Sub txtenm\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtenm.TextChanged

Dim empnm As String

empnm = txtenm.Text

For Each c As Char In empnm

If IsNumeric(c) = True Then

MessageBox.Show("NAME can't be accepted!!!!!" & vbCrLf & "Enter

Only Characters!!!!")

txtenm.Clear()

End If

Next

For Each c As Char In empnm

If Char.IsPunctuation(c) = True Then

MessageBox.Show("NAME can't be accepted!!!!!" & vbCrLf & "Enter Only Characters!!!!")

txtenm.Clear()

End If

Next

For Each c As Char In empnm

If Char.IsSymbol(c) = True Then

MessageBox.Show("NAME can't be accepted!!!!!" & vbCrLf & "Enter Only Characters!!!!")

txtenm.Clear()

End If

Next

End Sub

Private Sub txtphno\_leave(ByVal sender As System.Object, ByVal e As System.EventArgs)Handles txtphno.Leave

Dim phno As String

phno = txtphno.Text

Dim flag As Boolean

flag = True

For Each c As Char In phno

If Not IsNumeric(c) = True Then

flag = False

End If

Next

If flag = False Then

MessageBox.Show("Please Enter Only Numbers!!!!")

txtphno.Text = ""

ElseIf phno.Length < 8 Or phno.Length = 9 Or phno.Length >10

Then

MsgBox(" Phone Number Should Be Of Only 10 Digits!!!!")

txtphno.Text = ""

End If

End Sub

Private Sub txtsal\_TextChanged(ByVal sender As System.Object,

ByVal e As System.EventArgs) Handles txtsal.TextChanged

Dim flag As Boolean

flag = True

Dim salary As String

salary = txtsal.Text

For Each c As Char In salary

If Not IsNumeric(c) = True Then

flag = False

End If

Next

If flag = False Then

MessageBox.Show("Please Enter Only Numbers!!!!")

txtsal.Text = ""

End If

End Sub

Private Sub btnrefrsh\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnrefrsh.Click

txtenm.Enabled = False

txtphno.Enabled = True

rchad.Enabled = True

txtsal.Enabled = True

btndelete.Enabled = True

btnupdate.Enabled = True

End Sub

End Class

**Customer.vb**

Imports System.Data.SqlClient

Public Class Customer

Dim objCon As New SqlConnection("Data Source=.;Initial Catalog=gasagency;Integrated Security=True")

Dim objDataAdapter As New SqlDataAdapter("select \* from Customer", objCon)

Dim objDataSet As DataSet

Dim objDataView As DataView

Dim rowno As Integer

Dim objCurrencyManager As CurrencyManager

Private Sub fillDataSetAndView()

objDataSet = New DataSet()

objDataAdapter.Fill(objDataSet, "Customer")

objDataView = New DataView(objDataSet.Tables("Customer"))

objCurrencyManager = CType(Me.BindingContext(objDataView), CurrencyManager)

End Sub

Private Sub bindFields()

txtcid.DataBindings.Clear()

txtfnm.DataBindings.Clear()

txtlnm.DataBindings.Clear()

txtphno.DataBindings.Clear()

rchad.DataBindings.Clear()

cbonocyl.DataBindings.Clear()

txtcid.DataBindings.Add("Text", objDataView, "CustId")

txtfnm.DataBindings.Add("Text", objDataView, "CustFNm")

txtlnm.DataBindings.Add("Text", objDataView, "CustLNm")

txtphno.DataBindings.Add("Text", objDataView, "CustContact")

rchad.DataBindings.Add("Text", objDataView, "CustAdd")

cbonocyl.DataBindings.Add("Text", objDataView, "NoOfCyl")

End Sub

'---------------------following code for---------------

Grid View

'--------------------------------------------------------

Public Sub viewrecord()

objCon New SqlConnection("Data Source=.;Initial Catalog=gasagency;Integrated Security=True")

objCon.Open()

objDataAdapter = New SqlDataAdapter("select\* from Customer",objCon)

objDataSet = New DataSet

objDataAdapter.Fill(objDataSet, "Customer")

DataGridView1.DataSource = objDataSet

DataGridView1.DataMember = "Customer"

objCon.Close()

End Sub

Private Sub showPosition()

txtRecord.Text = objCurrencyManager.Position + 1 & " of " &

objCurrencyManager.Count()

End Sub

Private Sub Customer\_Load (ByVal sender As System.Object, ByVal e As System.EventArgs)MyBase.Load

fillDataSetAndView()

bindFields()

showPosition()

viewrecord()

txtcid.Enabled = False

txtfnm.Enabled = False

rchad.Enabled = False

txtphno.Enabled = False

txtlnm.Enabled = False

cbonocyl.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

DataGridView1.ReadOnly = True

End Sub

Private Sub btnnew\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnnew.Click

txtcid.Enabled = False

txtphno.Enabled = True

txtfnm.Enabled = True

txtlnm.Enabled = True

rchad.Enabled = True

rchad.Enabled = True

cbonocyl.Enabled = True

btnadd.Enabled = True

bindFields()

txtcid.Text = ""

'---------------------following code for-------------------

Automatically Generate Customer ID

'-----------------------------------------------------------

Dim strId As String

Dim MaxID As String = GetMaxID()

If MaxID = "" Then

txtcid.Text = "C101"

Else

strId = CType(MaxID.Remove(0, 1), String)

MaxID = CType(strId, Integer)

MaxID += 1

strId = "C" & MaxID.ToString

txtcid.Text = strId

End If

txtfnm.Text = ""

txtlnm.Text = ""

txtphno.Text = ""

rchad.Text = ""

cbonocyl.Text = ""

End Sub

Public Function GetMaxID() As String

Dim ds As DataSet = New DataSet

Dim objCon As New SqlConnection("Data Source=.;Initial

Catalog=gasagency;Integrated Security=True")

Dim cmd As SqlCommand = New SqlCommand("select MAX(CustId)from Customer", objCon)

objCon.Open()

Dim i As String = cmd.ExecuteScalar().ToString()

Return i

End Function

Private Sub btnfrst\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnfrst.Click

fillDataSetAndView()

bindFields()

objCurrencyManager.Position = 0

showPosition()

txtcid.Enabled = False

txtphno.Enabled = False

txtfnm.Enabled = False

txtlnm.Enabled = False

rchad.Enabled = False

rchad.Enabled = False

cbonocyl.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnnxt\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnnxt.Click

objCurrencyManager.Position += 1

showPosition()

txtcid.Enabled = False

txtphno.Enabled = False

txtfnm.Enabled = False

txtlnm.Enabled = False

rchad.Enabled = False

rchad.Enabled = False

cbonocyl.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnprev\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnprev.Click

objCurrencyManager.Position -= 1

showPosition()

txtcid.Enabled = False

txtphno.Enabled = False

txtfnm.Enabled = False

txtlnm.Enabled = False

rchad.Enabled = False

rchad.Enabled = False

cbonocyl.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnlast\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnlast.Click

fillDataSetAndView()

bindFields()

objCurrencyManager.Position = objCurrencyManager.Count() - 1

showPosition()

txtcid.Enabled = False

txtphno.Enabled = False

txtfnm.Enabled = False

txtlnm.Enabled = False

rchad.Enabled = False

rchad.Enabled = False

cbonocyl.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnsrch\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnsrch.Click

Dim i As Integer

objDataView.Sort = "CustId"

i = objDataView.Find(txtsrch.Text)

showPosition()

If (i = -1) Then

MessageBox.Show("SORRY!!Record not found!!!")

txtsrch.Text = ""

Else

objCurrencyManager.Position = i

txtsrch.Text = ""

End If

showPosition()

End Sub

Private Sub btnadd\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnadd.Click

Dim objCommand As SqlCommand = New SqlCommand

Dim i As Integer

i = objCurrencyManager.Position

If txtfnm.Text.Trim() = String.Empty Then

MessageBox.Show("Please enter Customre Name!!!!!")

ElseIf txtlnm.Text.Trim() = String.Empty Then

MessageBox.Show("Please enter Customer Last Name!!!!!")

ElseIf txtphno.Text.Trim() = String.Empty Then

MessageBox.Show("Please enter Customer Phone Number!!!!!")

ElseIf rchad.Text.Trim() = String.Empty Then

MessageBox.Show("Please enter Customer Address!!!!!")

ElseIf cbonocyl.Text.Trim() = String.Empty Then

MessageBox.Show("Please enter Number of cylinder customer have!!!!!")

Else

objCon.Open()

objCommand.Connection = objCon

objCommand.CommandText ="insert into Customer(CustId,

CustFnm,CustLnm,CustContact,CustAdd,NoOfCyl)Values((@CustId,@CustFnm,@CustLnm,@CustContact,@CustAdd,@NoOfCyl)"

MessageBox.Show("The record has been added successfully!!!!!")

objCommand.Parameters.AddWithValue("@CustId", txtcid.Text)

objCommand.Parameters.AddWithValue("@CustFnm", txtfnm.Text)

objCommand.Parameters.AddWithValue("@CustLnm", txtlnm.Text)

objCommand.Parameters.AddWithValue("@CustContact", txtphno.Text)

objCommand.Parameters.AddWithValue("@CustAdd", rchad.Text)

objCommand.Parameters.AddWithValue("@NoOfCyl", cbonocyl.Text)

Try

objCommand.ExecuteNonQuery()

Catch sqlExceptionErr As SqlException

MessageBox.Show(sqlExceptionErr.Message)

End Try

objCon.Close()

fillDataSetAndView()

bindFields()

showPosition()

objDataAdapter.Fill(objDataSet)

objCurrencyManager.Position = i

viewrecord()

txtphno.Enabled = False

txtfnm.Enabled = False

txtlnm.Enabled = False

rchad.Enabled = False

rchad.Enabled = False

cbonocyl.Enabled = False

btndelete.Enabled = False

btnupdate.Enabled = False

End If

End Sub

Private Sub btnupdate\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnupdate.Click

Dim objCommand As SqlCommand = New SqlCommand

objCon.Open()

objCommand.Connection = objCon

Dim i As Integer

i = objCurrencyManager.Position

objCommand.CommandText = "update Customer set CustAdd=@CusAdd, CustContact=@CustContact,NoOfCyl=@NoOfCyl whereCustId=@CustId"

MessageBox.Show("The record has been updated Sccessfully!!!!")

objCommand.Parameters.AddWithValue("@CustId", txtcid.Text)

objCommand.Parameters.AddWithValue("@CustFnm", txtfnm.Text)

objCommand.Parameters.AddWithValue("@CustLnm", txtlnm.Text)

objCommand.Parameters.AddWithValue("@CustContact", txtphno.Text)

objCommand.Parameters.AddWithValue("@CustAdd", rchad.Text)

objCommand.Parameters.AddWithValue("@NoOfCyl", cbonocyl.Text)

Try

objCommand.ExecuteNonQuery()

Catch sqlExceptionErr As SqlException

MessageBox.Show(sqlExceptionErr.Message)

End Try

objCon.Close()

fillDataSetAndView()

bindFields()

objCurrencyManager.Position = i

showPosition()

viewrecord()

txtphno.Enabled = False

txtfnm.Enabled = False

txtlnm.Enabled = False

rchad.Enabled = False

rchad.Enabled = False

cbonocyl.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

End Sub

Private Sub btndelete\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btndelete.Click

Dim intposition As Integer

Dim objCommamnd As SqlCommand = New SqlCommand()

intposition = Me.BindingContext(objDataView).Position - 1

If intposition < 0 Then

intposition = 0

End If

Dim x = MsgBox("Are You Sure you want to delete this record",

MsgBoxStyle.Question + MsgBoxStyle.YesNo)

If x = vbYes Then

objCommamnd.Connection = objCon

objCommamnd.CommandText = "delete from Customer where custId=@CustId"

objCommamnd.Parameters.AddWithValue("@CustId",BindingContext( objDataView).(Current("CustId"))

MessageBox.Show("The record has been deleted Sccessfully!!")

objCon.Open()

objCommamnd.ExecuteNonQuery()

txtcid.Text = ""

txtfnm.Text = ""

txtlnm.Text = ""

txtphno.Text = ""

rchad.Text = ""

cbonocyl.Text = ""

objCon.Close()

fillDataSetAndView()

bindFields()

viewrecord()

Me.BindingContext(objDataView).Position = intposition

showPosition()

txtfnm.Enabled = False

txtphno.Enabled = False

rchad.Enabled = False

txtlnm.Enabled = False

cbonocyl.Enabled = False

btnupdate.Enabled = False

btnadd.Enabled = False

EndIf

End Sub

Private Sub btnexit\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnexit.Click

Me.Close()

End Sub

Private Sub txtfnm\_TextChanged(ByVal sender As System.Object,

ByVal e As System.EventArgs) Handles txtfnm.TextChanged

Dim custnm As String

custnm = txtfnm.Text

For Each c As Char In custnm

If IsNumeric(c) = True Then

MessageBox.Show("NAME can't be accepted!!!!!" & vbCrLf & "Enter Only Characters!!!!")

txtfnm.Clear()

End If

Next

For Each c As Char In custnm

If Char.IsPunctuation(c) = True Then

MessageBox.Show("NAME can't be accepted!!!!!" & vbCrLf & "Enter Only Characters!!!!")

txtfnm.Clear()

End If

Next

For Each c As Char In custnm

If Char.IsSymbol(c) = True Then

MessageBox.Show("NAME can't be accepted!!!!!" & vbCrLf & "Enter Only Characters!!!!")

txtfnm.Clear()

End If

Next

End Sub

Private Sub txtlnm\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtlnm.TextChanged

Dim custlnm As String

custlnm = txtlnm.Text

For Each c As Char In custlnm

If IsNumeric(c) = True Then

MessageBox.Show(" LAST NAME can't be accepted!!!!!" & vbCrLf & "Enter Only Characters!!!!")

txtlnm.Text = ""

End If

Next

For Each c As Char In custlnm

If Char.IsPunctuation(c) = True Then

MessageBox.Show("LAST NAME can't be accepted!!!!!" vbCrLf & "Enter Only Characters!!!!")

txtlnm.Clear()

End If

Next

For Each c As Char In custlnm

If Char.IsSymbol(c) = True Then

MessageBox.Show("LAST NAME can't be accepted!!!!!" & vbCrLf & "Enter Only Characters!!!!")

txtlnm.Clear()

End If

Next

End Sub

Private Sub txtphno\_Leave(ByVal sender As Object, ByVal e

As System.EventArgs) Handles txtphno.Leave

Dim phno As String

phno = txtphno.Text

Dim flag As Boolean

flag = True

For Each c As Char In phno

If Not IsNumeric(c) = True Then

flag = False

End If

Next

For Each c As Char In phno

If Char.IsPunctuation(c) = True Then

flag = False

End If

Next

For Each c As Char In phno

If Char.IsSymbol(c) = True Then

flag = False

End If

Next

If flag = False Then

MessageBox.Show("Please Enter Only Numbers!!!!")

txtphno.Text = ""

ElseIf phno.Length < 8 Or phno.Length = 9 Or phno.Length > 10

Then

MsgBox(" Phone Number Should Be Of Only 10 Digits!!!!")

txtphno.Text = ""

End If

End Sub

Private Sub btnrefrsh\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnrefrsh.Click

txtphno.Enabled = True

txtfnm.Enabled = False

txtlnm.Enabled = False

rchad.Enabled = True

rchad.Enabled = True

cbonocyl.Enabled = True

btndelete.Enabled = True

btnupdate.Enabled = True

End Sub

End Class

**Supplier.vb**

Imports System.Data.SqlClient

Public Class Supplier

Dim objCon As New SqlConnection("Data Source=.;Initial Catalog=gasagency;Integrated Security=True")

Dim objDataAdapter As New SqlDataAdapter("select \* from Supplier", objCon)

Dim objDataSet As DataSet

Dim objDataView As DataView

Dim objCurrencyManager As CurrencyManager

Private Sub fillDataSetAndView()

objDataSet = New DataSet()

objDataAdapter.Fill(objDataSet, "Supplier")

objDataView = New DataView(objDataSet.Tables("Supplier"))

objCurrencyManager = CType(Me.BindingContext(objDataView), CurrencyManager)

End Sub

Private Sub bindFields()

txtsid.DataBindings.Clear()

txtsnm.DataBindings.Clear()

txtcontno.DataBindings.Clear()

rchad.DataBindings.Clear()

txtsid.DataBindings.Add("Text", objDataView, "SupId")

txtsnm.DataBindings.Add("Text", objDataView, "SupNm")

txtcontno.DataBindings.Add("Text", objDataView, "SupContact")

rchad.DataBindings.Add("Text", objDataView, "SupAdd")

End Sub

'---------------------Following code for--------------------------

Grid View

'------------------------------------------------------------------

Public Sub viewrecord()

objCon = New SqlConnection("Data Source=.;Initial

Catalog=gasagency;

Integrated Security=True")

objCon.Open()

objDataAdapter = New SqlDataAdapter("select\* from Supplier", objCon)

objDataSet = New DataSet

objDataAdapter.Fill(objDataSet, "Supplier")

DataGridView1.DataSource = objDataSet

DataGridView1.DataMember = "Supplier"

objCon.Close()

End Sub

Private Sub showPosition()

txtRecord.Text = objCurrencyManager.Position + 1 & " of " &

objCurrencyManager.Count()

End Sub

Private Sub Supplier\_Load (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles MyBase.Load

fillDataSetAndView()

bindFields()

showPosition()

viewrecord()

txtsid.Enabled = False

txtsnm.Enabled = False

rchad.Enabled = False

txtcontno.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

DataGridView1.ReadOnly = True

End Sub

Private Sub btnfrst\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnfrst.Click

fillDataSetAndView()

bindFields()

objCurrencyManager.Position = 0

showPosition()

txtsid.Enabled = False

txtsnm.Enabled = False

txtcontno.Enabled = False

rchad.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnnxt\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnnxt.Click

objCurrencyManager.Position += 1

showPosition()

txtsid.Enabled = False

txtsnm.Enabled = False

txtcontno.Enabled = False

rchad.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnprev\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnprev.Click

objCurrencyManager.Position -= 1

showPosition()

txtsid.Enabled = False

txtsnm.Enabled = False

txtcontno.Enabled = False

rchad.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnlast\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnlast.Click

fillDataSetAndView()

bindFields()

objCurrencyManager.Position = objCurrencyManager.Count() - 1

showPosition()

txtsid.Enabled = False

txtsnm.Enabled = False

txtcontno.Enabled = False

rchad.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

btnupdate.Enabled = False

End Sub

Private Sub btnsrch\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnsrch.Click

Dim i As Integer

objDataView.Sort = "SupId"

i = objDataView.Find(txtsrch.Text)

showPosition()

If (i = -1) Then

MessageBox.Show("SORRY!!Record not found!!!")

txtsrch.Text = ""

Else

objCurrencyManager.Position = i

txtsrch.Text = ""

End If

showPosition()

End Sub

Private Sub btnnew\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnnew.Click

txtsid.Enabled = False

bindFields()

txtsid.Text = ""

'---------------------Following code for-------------------

Automatically Generate Supplier ID

'----------------------------------------------------------

Dim strId As String

Dim MaxID As String = GetMaxID()

If MaxID = "" Then

txtsid.Text = "S101"

Else

strId = CType(MaxID.Remove(0, 1), String)

MaxID = CType(strId, Integer)

MaxID += 1

strId = "S" & MaxID.ToString

txtsid.Text = strId

End If

txtsnm.Text = ""

txtcontno.Text = ""

rchad.Text = ""

viewrecord()

txtcontno.Enabled = True

txtsnm.Enabled = True

rchad.Enabled = True

btnadd.Enabled = True

End Sub

Public Function GetMaxID() As String

Dim ds As DataSet = New DataSet

Dim objCon As New SqlConnection("Data Source=.;Initial Catalog=gasagency;Integrated

Security=True")

Dim cmd As SqlCommand = New SqlCommand("select MAX(SupId)from Supplier",objCon)

objCon.Open()

Dim i As String = cmd.ExecuteScalar().ToString()

Return i

End Function

Private Sub btnadd\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnadd.Click

Dim objCommand As SqlCommand = New SqlCommand

Dim i As Integer

If txtsnm.Text.Trim() = String.Empty Then

MessageBox.Show("Please enter Supplier Name!!!!!")

ElseIf txtcontno.Text.Trim() = String.Empty Then

MessageBox.Show("Please enter Customer Phone Number!!!!!")

ElseIf rchad.Text.Trim() = String.Empty Then

MessageBox.Show("Please enter Supplier Address!!!!!")

Else

objCon.Open()

objCommand.Connection = objCon

i = objCurrencyManager.Position

objCommand.CommandText = "insert into Supplier(SupId,SupNm,SupContact,SupAdd)Values(@SupId,@SupNm,@SupContact,@SupAdd)”

MessageBox.Show("The record has been added successfully!")

objCommand.Parameters.AddWithValue("@SupId", txtsid.Text)

objCommand.Parameters.AddWithValue("@SupNm", txtsnm.Text)

ly!! been added successfully!!!,objCommand.Parameters.AddWithValue("@SupContact", txtcontno.Text)

objCommand.Parameters.AddWithValue("@SupAdd", rchad.Text)

Try

objCommand.ExecuteNonQuery()

Catch sqlExceptionErr As SqlException

MessageBox.Show(sqlExceptionErr.Message)

End Try

objCon.Close()

fillDataSetAndView()

bindFields()

objCurrencyManager.Position = i

showPosition()

viewrecord()

txtsnm.Enabled = False

txtcontno.Enabled = False

rchad.Enabled = False

btndelete.Enabled = False

btnupdate.Enabled = False

End If

End Sub

Private Sub btnupdate\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnupdate.Click

Dim objCommand As SqlCommand = New SqlCommand

objCon.Open()

objCommand.Connection = objCon

Dim i As Integer

i = objCurrencyManager.Position

objCommand.CommandText = "update Supplier set SupContact=

@SupContact, SupAdd=@SupAdd where SupId=@SupId"

MessageBox.Show("The record has been updated successfully! ")

objCommand.Parameters.AddWithValue("@SupId", txtsid.Text)

objCommand.Parameters.AddWithValue("@SupNm", txtsnm.Text)

objCommand.Parameters.AddWithValue("@SupContact", txtcontno.Text)

objCommand.Parameters.AddWithValue("@SupAdd", rchad.Text)

Try

objCommand.ExecuteNonQuery()

Catch sqlExceptionErr As SqlException

MessageBox.Show(sqlExceptionErr.Message)

End Try

objCon.Close()

fillDataSetAndView()

bindFields()

objCurrencyManager.Position = i

showPosition()

viewrecord()

txtsnm.Enabled = False

txtcontno.Enabled = False

rchad.Enabled = False

btndelete.Enabled = False

btnadd.Enabled = False

End Sub

Private Sub btndelete\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btndelete.Click

Dim intposition As Integer

Dim objCommamnd As SqlCommand = New SqlCommand()

intposition = Me.BindingContext(objDataView).Position - 1

If intposition < 0 Then

intposition = 0

End If

Dim x = MsgBox("Are You Sure you want to delete this record",

MsgBoxStyle. Question + MsgBoxStyle.YesNo)

If x = vbYes Then

objCommamnd.Connection = objCon

objCommamnd.CommandText = "delete from Supplier where SupId=@SupId" objCommamnd.Parameters.AddWithValue("@SupId",BindingContext(objDataView).Current("SupId"))

MessageBox.Show("The record has been deleted successfully!!!!!!!")

objCon.Open()

objCommamnd.ExecuteNonQuery()

txtsid.Text = ""

txtsnm.Text = ""

txtcontno.Text = ""

rchad.Text = ""

objCon.Close()

fillDataSetAndView()

bindFields()

viewrecord()

Me.BindingContext(objDataView).Position = intposition

showPosition()

txtsnm.Enabled = False

txtcontno.Enabled = False

rchad.Enabled = False

btnadd.Enabled = False

btnadd.Enabled = False

EndIf

End Sub

Private Sub btnexit\_Click(ByVal sender As System.Object, ByVal e

As System.EventArgs) Handles btnexit.Click

Me.Close()

End Sub

Private Sub txtsnm\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles txtsnm.TextChanged

Dim supnm As String

supnm = txtsnm.Text

For Each c As Char In supnm

If IsNumeric(c) = True Then

MessageBox.Show("NAME can't be accepted!!!!!" & vbCrLf & "Enter Only Characters!!!!")

txtsnm.Clear()

End If

Next

For Each c As Char In supnm

If Char.IsPunctuation(c) = True Then

MessageBox.Show("LAST NAME can't be accepted!!!!!" & vbCrLf & "Enter Only Characters!!!!")

txtsnm.Clear()

End If

Next

For Each c As Char In supnm

If Char.IsSymbol(c) = True Then

MessageBox.Show("LAST NAME can't be accepted!!!!!" & vbCrLf & "Enter Only

Characters!!!!")

txtsnm.Clear()

End If

Next

End Sub

Private Sub txtcontno\_Leave (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles txtcontno.Leave

Dim no As String

no = txtcontno.Text

Dim flag As Boolean

flag = True

Dim phno As String

phno = txtcontno.Text

For Each c As Char In phno

If Not IsNumeric(c) = True Then

flag = False

End If

Next

For Each c As Char In phno

If Char.IsPunctuation(c) = True Then

flag = False

End If

Next

For Each c As Char In phno

If Char.IsSymbol(c) = True Then

flag = False

End If

Next

If flag = False Then

MessageBox.Show("Please Enter Only Numbers!!!!")

txtcontno.Text = ""

ElseIf no.Length < 8 Or no.Length = 9 Or no.Length > 10

Then

MsgBox(" Phone Number Should Be Of Only 10 Digits!!!!")

txtcontno.Text = ""

End If

End Sub

Private Sub btnrefrsh\_Click (ByVal sender As System.Object, ByVal e As System.EventArgs)Handles btnrefrsh.Click

txtsnm.Enabled = False

txtcontno.Enabled = True

rchad.Enabled = True

btnupdate.Enabled = True

btndelete.Enabled = True

End Sub

End Class

**Factory Pattern**

Factory pattern is one of the most used design patterns in Java. This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object.

In Factory pattern, we create object without exposing the creation logic to the client and refer to newly created object using a common interface.

Q 2. Identify at least **two GRASP Design Patterns** that can be used. Provide details of how these patterns can be applied in order to improve the Object-Oriented design of the system.

General Responsibility Assignment Software Patterns (or Principles), abbreviated GRASP, consist of guidelines for assigning responsibility to classes and objects in object-oriented design. It is not related to the SOLID design principle.

The different patterns and principles used in GRASP are controller, creator, indirection, information expert, high cohesion, low coupling, polymorphism, protected variations, and pure fabrication. All these patterns answer some software problem, and these problems are common to almost every software development project. These techniques have not been invented to create new ways of working, but to better document and standardize old, tried-and-tested programming principles in object-oriented design.

Patterns

In OO design, a pattern is a named description of a problem and solution that can be applied to new contexts; ideally, a pattern advises us on how to apply its solution in varying circumstances and considers the forces and trade-offs. Many patterns, given a specific category of problem, guide the assignment of responsibilities to objects.

1. Controller

The controller pattern assigns the responsibility of dealing with system events to a non- UI class that represents the overall system or a use case scenario. A controller object is a non-user interface object responsible for receiving or handling a system event.

A use case controller should be used to deal with all system events of a use case, and may be used for more than one use case (for instance, for use cases Create User and Delete User, one can have a single class called UserController, instead of two separate use case controllers(This may save some space)).

The controller is defined as the first object beyond the UI layer that receives and coordinates ("controls") a system operation. The controller should delegate the work that needs to be done to other objects; it coordinates or controls the activity. It should not do much work itself. The GRASP Controller can be thought of as being a part of the application/service layer [2] (assuming that the application has made an explicit distinction between the application/service layer and the domain layer) in an object-oriented system with common layers in an information system logical architecture.

1. Creator

Creation of objects is one of the most common activities in an object-oriented system. Which class is responsible for creating objects is a fundamental property of the relationship between objects of particular classes.In general, a class B should be responsible for creating instances of class A if one, or preferably more, of the following apply:

Instances of B contain or compositely aggregate instances of A

Instances of B record instances of A

Instances of B closely use instances of A

Instances of B have the initializing information for instances of A and pass it on creation.

1. High cohesion

High cohesion is an evaluative pattern that attempts to keep objects appropriately focused, manageable and understandable. High cohesion is generally used in support of low coupling. High cohesion means that the responsibilities of a given element are strongly related and highly focused. Breaking programs into classes and subsystems is an example of activities that increase the cohesive properties of a system. Alternatively, low cohesion is a situation in which a given element has too many unrelated responsibilities. Elements with low cohesion often suffer from being hard to comprehend, hard to reuse, hard to maintain and averse to change.[3]

1. Information expert

Information expert (also expert or the expert principle) is a principle used to determine where to delegate responsibilities. These responsibilities include methods, computed fields, and so on.Using the principle of information expert, a general approach to assigning responsibilities is to look at a given responsibility, determine the information needed to fulfill it, and then determine where that information is stored.

1. Low coupling

Coupling is a measure of how strongly one element is connected to, has knowledge of, or relies on other elements. Low coupling is an evaluative pattern that dictates how to assign responsibilities to support lower dependency between the classes, change in one class having lower impact on other classes, higher reuse potential.

1. Polymorphism

According to polymorphism principle, responsibility of defining the variation of behaviors based on type is assigned to the type for which this variation happens. This is achieved using polymorphic operations. The user of the type should use polymorphic operations instead of explicit branching based on type.

1. Protected variations

The protected variations pattern protects elements from the variations on other elements (objects, systems, subsystems) by wrapping the focus of instability with an interface and using polymorphism to create various implementations of this interface.

1. Pure fabrication

A pure fabrication is a class that does not represent a concept in the problem domain, specially made up to achieve low coupling, high cohesion, and the reuse potential thereof derived (when a solution presented by the information expert pattern does not). This kind of class is called a "service" in domain-driven design.