GHANA COMMUNICATION TECHNOLOGY UNIVERSITY



FACULTY OF COMPUTING AND INFORMATION SYSTEMS

DEPARTMENT OF INFORMATION TECHNOLOGY

CICS 314: ADVANCE VISUAL BASICS PROGRAMMING.NET

ASSIGNMENT 2

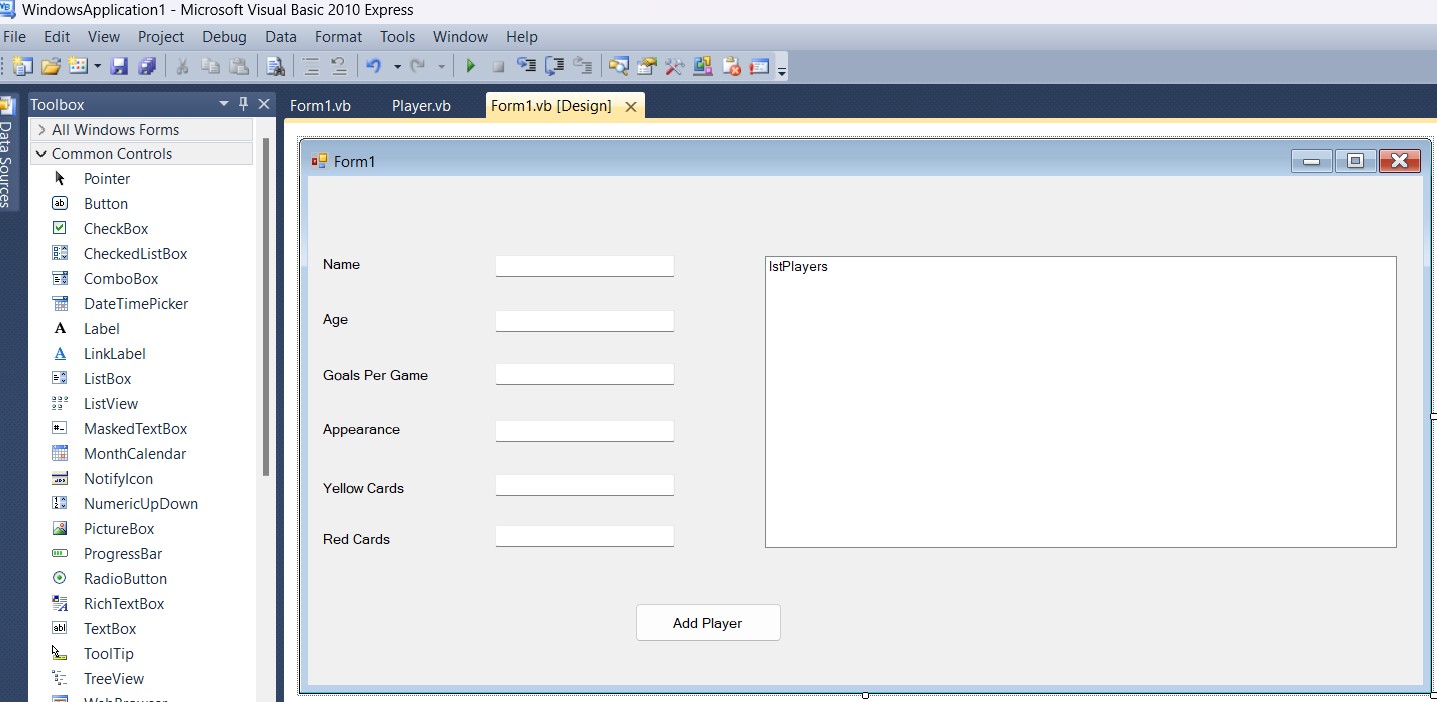
NAME: ELLIS ARMAH AYIKWEI

INDEX NUMBER: 4211220874

LEVEL: 300 EVENING

DATE: JULY 26, 2024

LECTURER: MRS ANANYA SANTOSH



Public Class Form1

Private players As New List(Of Player)()

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

Try

' Validate input data

Dim age As Integer = Convert.ToInt32(txtAge.Text)

Dim goalsPerGame As Double =

Convert.ToDouble(txtGoalsPerGame.Text)

Dim appearances As Integer =

Convert.ToInt32(txtAppearance.Text)

Dim yellowCards As Integer =

Convert.ToInt32(txtYellowCard.Text)

Dim redCards As Integer =

Convert.ToInt32(txtRedCard.Text)

' Create a new player object with validated data Dim player As New Player() player.Name = txtName.Text player.Age = age

player.GoalsPerGame = goalsPerGame player.Appearances = appearances player.YellowCards = yellowCards player.RedCards = redCards

' Add the player to the list and update the display players.Add(player) DisplayPlayers()

Catch ex As Exception

MessageBox.Show("Please enter valid data.", "Input

Error", MessageBoxButtons.OK, MessageBoxIcon.Error)

End Try

End Sub

Private Sub DisplayPlayers() lstPlayers.Items.Clear() For Each player In players

lstPlayers.Items.Add("Name: " & player.Name & ", Age: "

& player.Age & ", Goals/Game: " & player.GoalsPerGame & ", Appearances: " & player.Appearances & ", Yellow Cards: " & player.YellowCards & ", Red Cards: " & player.RedCards)

Next

End Sub

End Class

Public Class Player

Public Property Name As String

Public Property Age As Integer

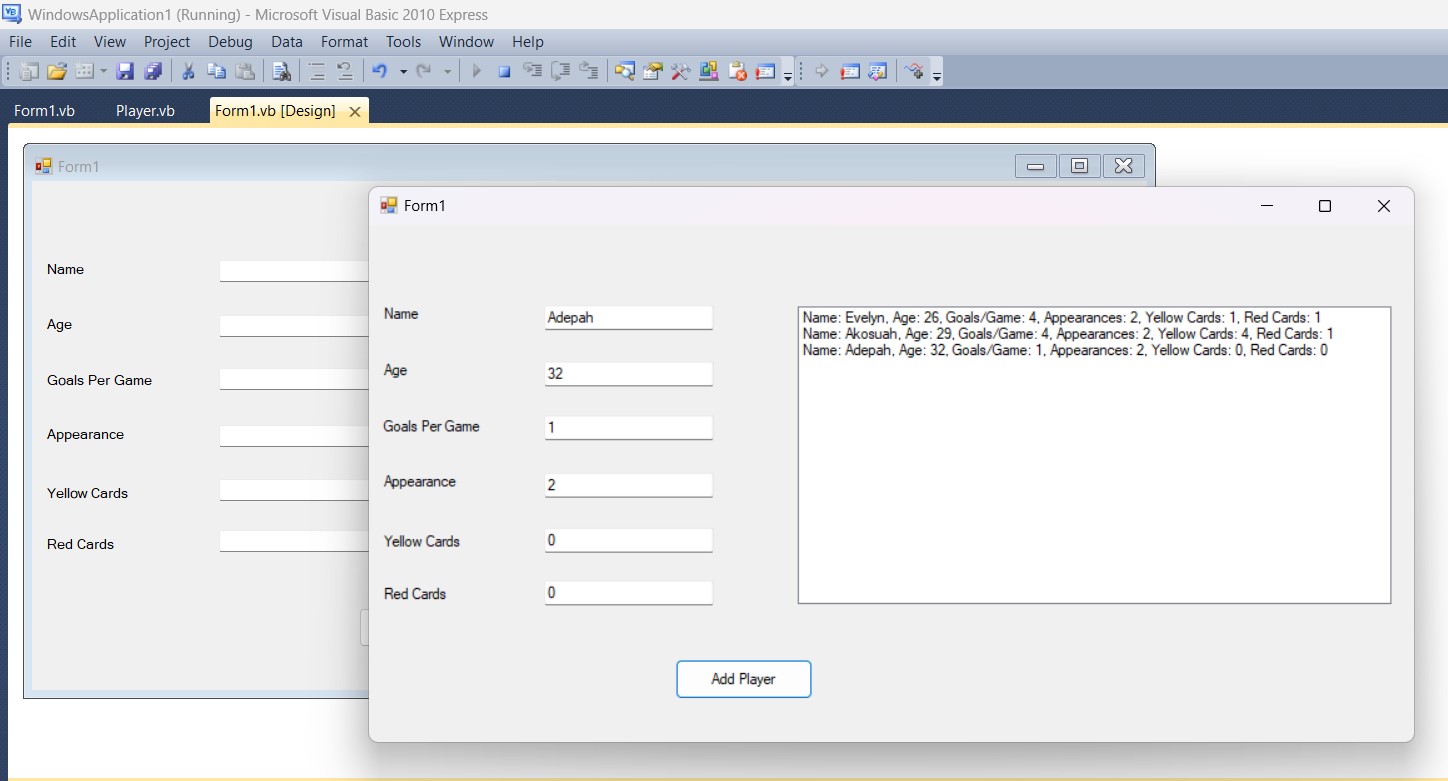
Public Property GoalsPerGame As Double

Public Property Appearances As Integer

Public Property YellowCards As Integer

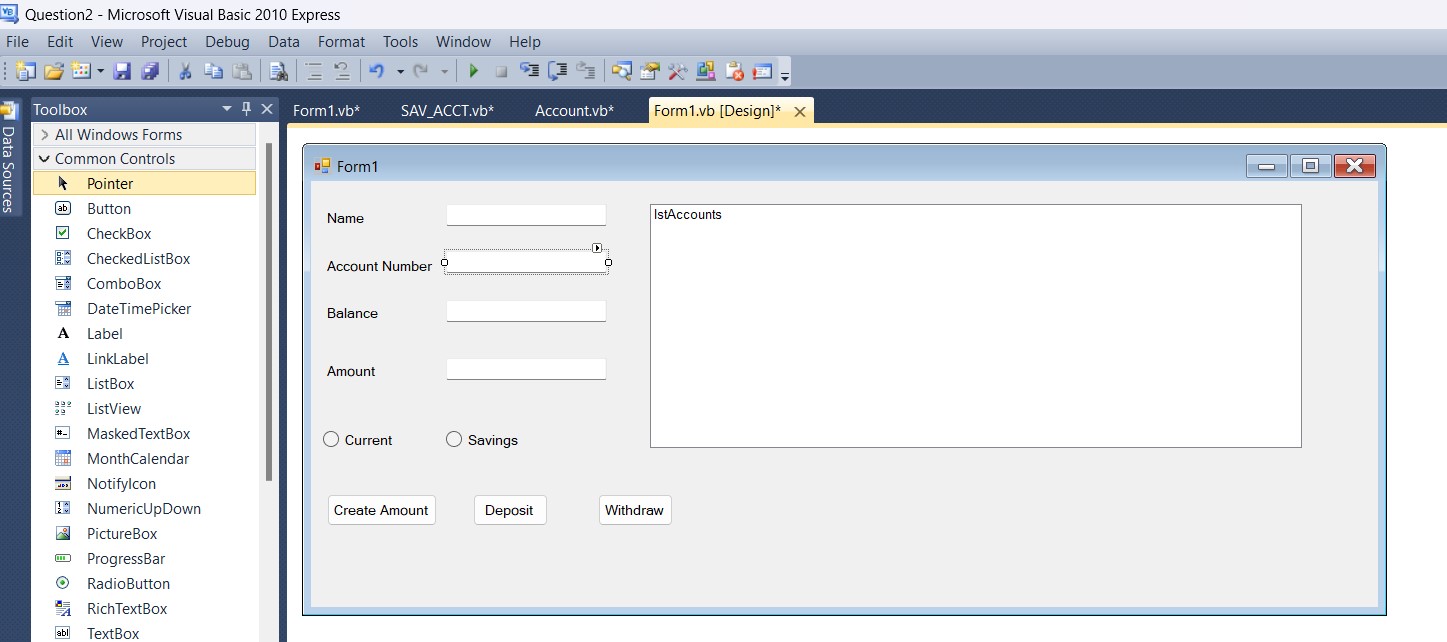
Public Property RedCards As Integer

End Class



**Result Screen**

# Question 2



Public Class Form1

Private accounts As New List(Of Account)()

Private Sub btnCreateAccount\_Click(ByVal sender As Object, ByVal

e As EventArgs) Handles btnCreateAccount.Click

Dim accType As String If rdoCurrent.Checked Then accType = "Current" Else

accType = "Savings"

End If

Dim account As Account If accType = "Current" Then

account = New CUR\_ACCT(txtName.Text, txtAccNum.Text,

Convert.ToDouble(txtBalance.Text)) Else

account = New SAV\_ACCT(txtName.Text, txtAccNum.Text,

Convert.ToDouble(txtBalance.Text))

End If

accounts.Add(account)

DisplayAccounts()

End Sub

Private Sub btnDeposit\_Click(ByVal sender As Object, ByVal e As

EventArgs) Handles btnDeposit.Click

Dim accNum As String = txtAccNum.Text

Dim amount As Double = Convert.ToDouble(txtAmount.Text)

Dim account As Account = FindAccountByNumber(accNum)

If account IsNot Nothing Then account.Deposit(amount)

MessageBox.Show("Deposit Successful")

DisplayAccounts()

Else

MessageBox.Show("Account not found.")

End If

End Sub

Private Sub btnWithdraw\_Click(ByVal sender As Object, ByVal e As

EventArgs) Handles btnWithdraw.Click

Dim accNum As String = txtAccNum.Text

Dim amount As Double = Convert.ToDouble(txtAmount.Text)

Dim account As Account = FindAccountByNumber(accNum)

If account IsNot Nothing Then

If account.Withdraw(amount) Then

MessageBox.Show("Withdrawal Successful")

Else

MessageBox.Show("Insufficient Balance")

End If

DisplayAccounts()

Else

MessageBox.Show("Account not found.")

End If

End Sub

Private Sub DisplayAccounts() lstAccounts.Items.Clear()

For Each account As Account In accounts

lstAccounts.Items.Add(account.AccountType & " Account -

" & account.CustomerName & ", Balance: " &

account.Balance.ToString())

Next

End Sub

Private Function FindAccountByNumber(ByVal accNum As String) As

Account

For Each account As Account In accounts

If account.AccountNumber = accNum Then

Return account

End If

Next

Return Nothing

End Function

End Class

Public Class CUR\_ACCT

Inherits Account

Private Const MinimumBalance As Double = 1000.0

Private Const Penalty As Double = 50.0

Public Sub New(ByVal name As String, ByVal accNum As String,

ByVal balance As Double)

MyBase.New(name, accNum, "Current", balance)

End Sub

Public Sub CheckMinimumBalance()

If Balance < MinimumBalance Then

Balance -= Penalty

MessageBox.Show(String.Format("Penalty of {0} imposed

due to insufficient balance.", Penalty))

End If

End Sub

End Class

Public Class SAV\_ACCT

Inherits Account

Private Const InterestRate As Double = 0.03

Public Sub New(ByVal name As String, ByVal accNum As String,

ByVal balance As Double)

MyBase.New(name, accNum, "Savings", balance)

End Sub

Public Sub ComputeInterest()

Dim interest As Double = Me.Balance \* InterestRate

Me.Deposit(interest)

End Sub

End Class

Public Class Account

Public Property CustomerName As String

Public Property AccountNumber As String

Public Property AccountType As String

Public Property Balance As Double

Public Sub New(ByVal name As String, ByVal accNum As String,

ByVal accType As String, ByVal balance As Double)

Me.CustomerName = name

Me.AccountNumber = accNum

Me.AccountType = accType

Me.Balance = balance

End Sub

Public Overridable Sub Deposit(ByVal amount As Double)

Me.Balance += amount

End Sub

Public Overridable Function Withdraw(ByVal amount As Double) As

Boolean

If amount <= Me.Balance Then

Me.Balance -= amount

Return True

End If

Return False

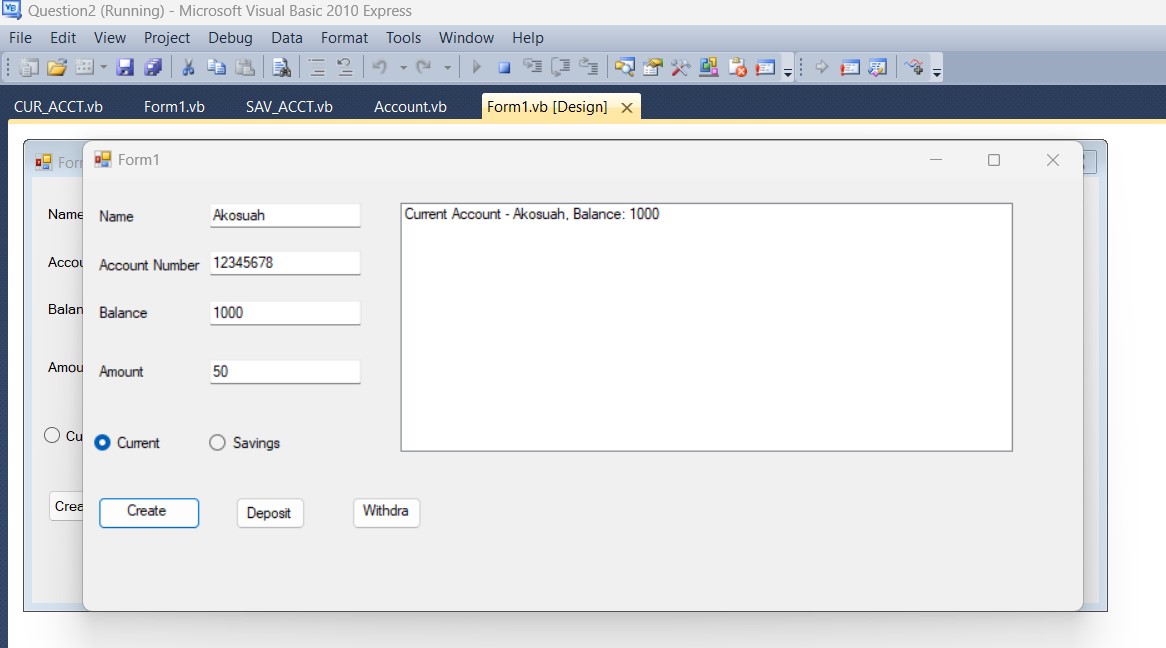
End Function

Public Sub DisplayBalance()

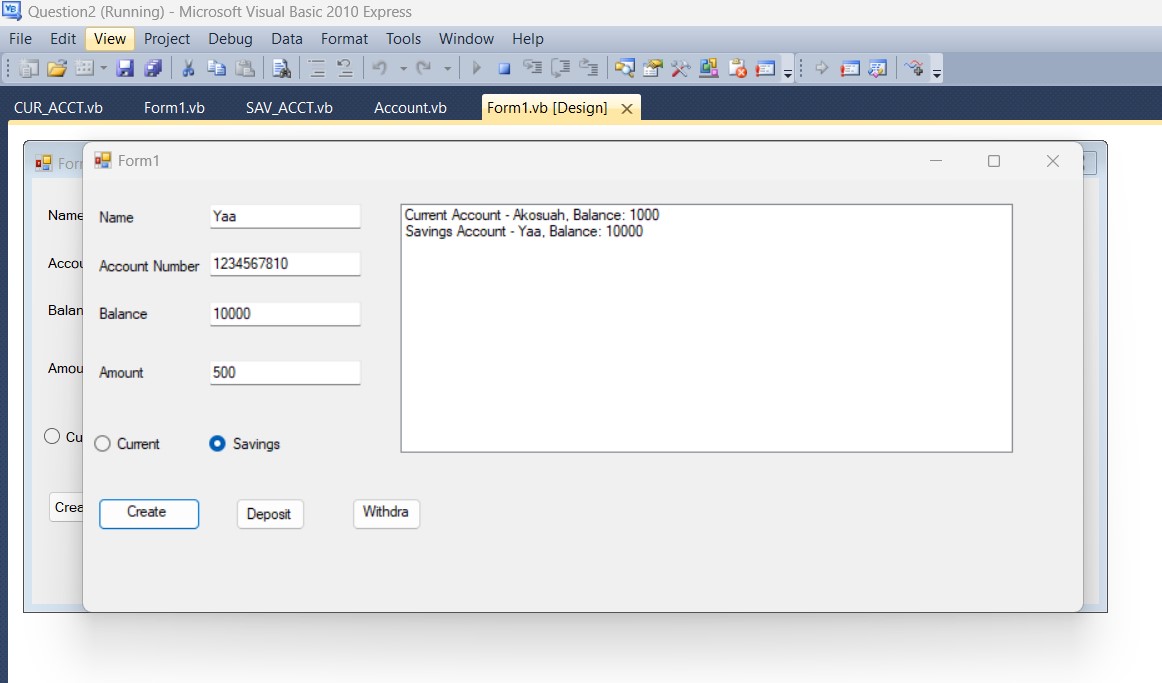
MessageBox.Show("Balance: " & Me.Balance.ToString())

End Sub

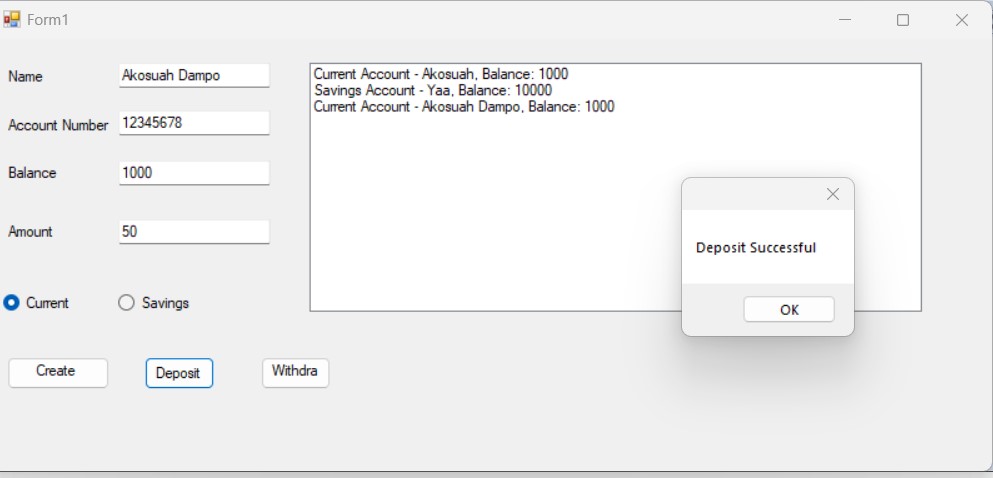
End Class



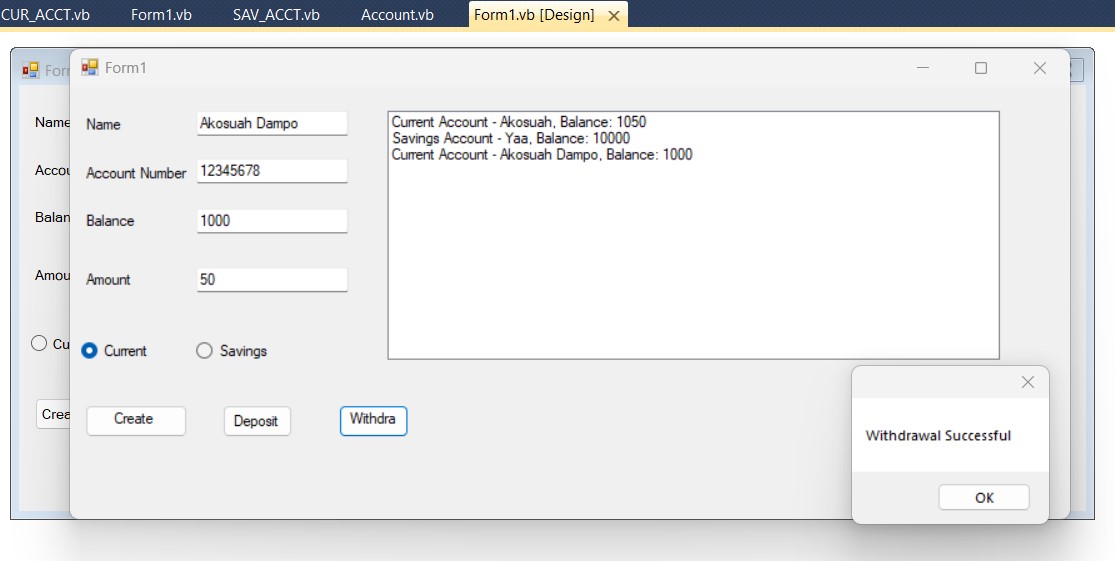
# Creating Current Account



# Creating Savings Account

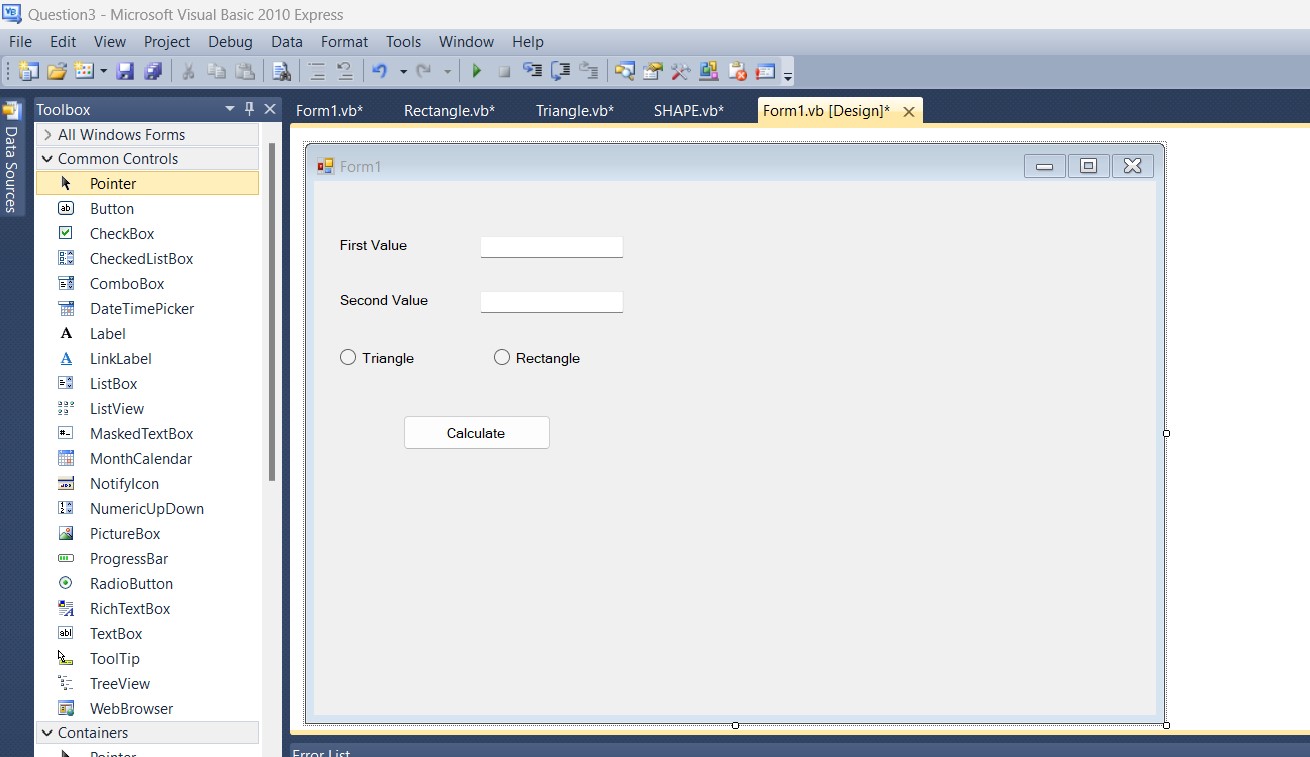


# Deposit



**Withdrawal**

# Question 3



Public Class SHAPE

Protected Value1 As Double

Protected Value2 As Double

Public Sub GetData(ByVal v1 As Double, ByVal v2 As Double)

Value1 = v1

Value2 = v2

End Sub

Public Overridable Sub DisplayArea()

MessageBox.Show("Area is not defined.")

End Sub

End Class

Public Class Form1

Private shape As SHAPE

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

Handles btnCalculate.Click

Dim v1 As Double = Convert.ToDouble(txtValue1.Text)

Dim v2 As Double = Convert.ToDouble(txtValue2.Text)

If rdoTriangle.Checked Then shape = New Triangle() Else

shape = New Rectangle()

End If

shape.GetData(v1, v2) shape.DisplayArea()

End Sub

End Class

Public Class Rectangle

Inherits SHAPE

Public Overrides Sub DisplayArea()

Dim area As Double = Value1 \* Value2

MessageBox.Show(String.Format("Area of Rectangle: {0}", area))

End Sub

End Class

Public Class Triangle

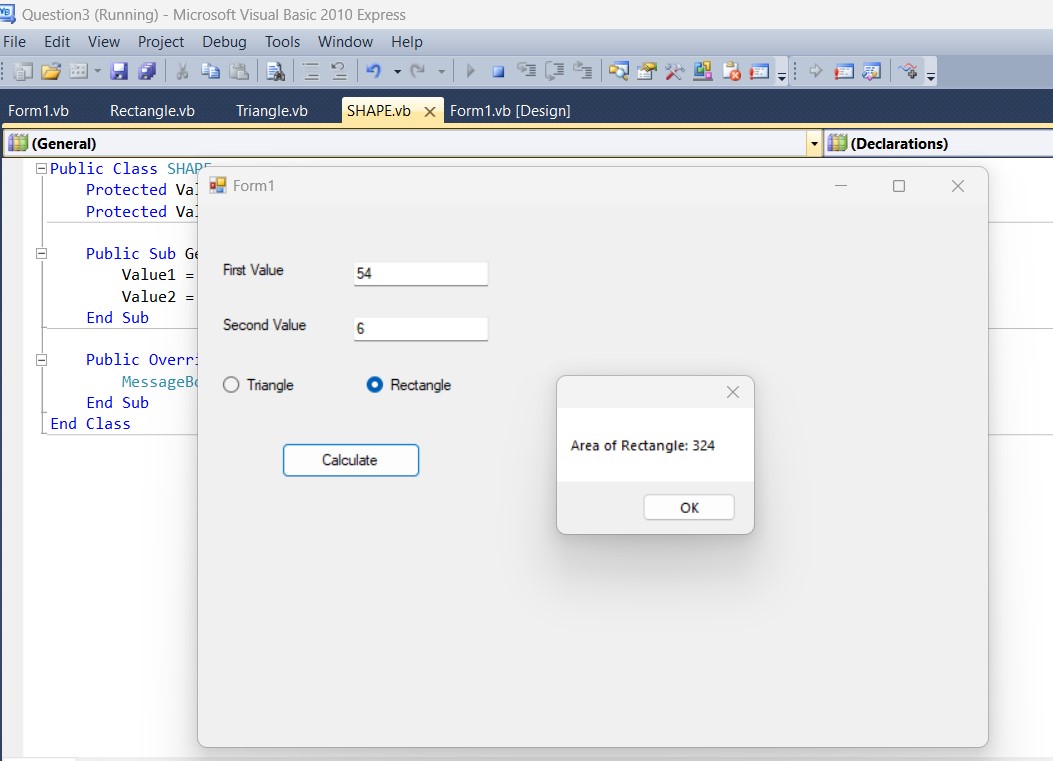
Inherits SHAPE

Public Overrides Sub DisplayArea()

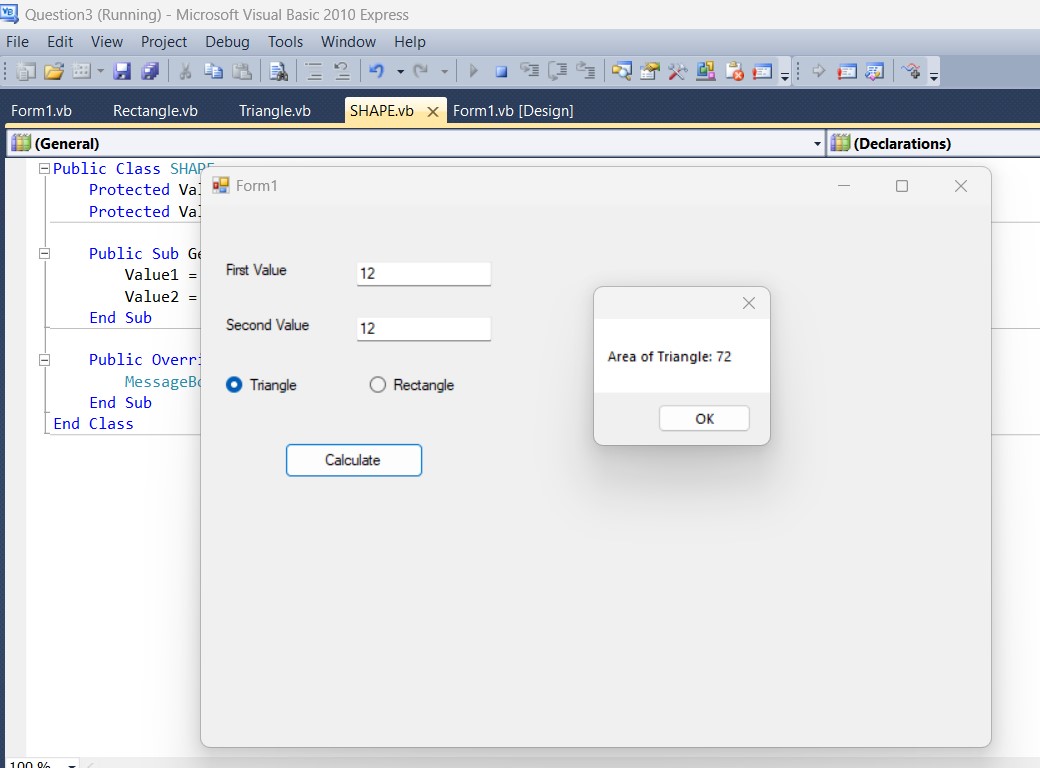
Dim area As Double = 0.5 \* Value1 \* Value2

MessageBox.Show(String.Format("Area of Triangle: {0}", area)) End Sub

End Class



# Rectangle



# Triangle