

```

' *****
' Surname, Initials:Kuni, D
' Student Number:201320596
' Practical: P11
' Class name: frmGDU
' *****
Option Strict On
Option Infer Off
Option Explicit On
Imports System.Runtime.Serialization.Formatters.Binary
Imports System.IO
Public Class frmGDU
#Region "Attributes"
    Private fs As FileStream
    Private bf As BinaryFormatter
    Private Members() As cMember
    Private NumberOfMembers As Integer
    Private Const filename As String = "S201320596.ifm"
#End Region
#Region "Question C"
    Private Sub btnSetup_Click(sender As Object, e As EventArgs) Handles btnSetup.Click
        'resize of array and gets number of ministers
        Try
            NumberOfMembers = CInt(InputBox("Please enter the number Of members "))
            ReDim Members(NumberOfMembers)
        Catch ex As Exception
        End Try
    End Sub
    Private Sub btnenterInfo_Click(sender As Object, e As EventArgs) Handles
btnenterInfo.Click
        'getting inputs from the user
        Dim choice As Integer

        Try
            choice = CInt(InputBox("Please enter the number 1 for representative or the
number 2 for Minister"))
            Select Case choice
            Case 1
                Dim Name As String, Reputation As Integer, Desiredbudget As Double,
Popularity As Integer, years As Integer
                Name = InputBox("Please enter the name of the representative member
")
                Reputation = CInt(InputBox("Please enter the reputation level of the
representative member (an Integer value between 1 and 10)"))
                Desiredbudget = CDb1(InputBox("Please enter the desired budget of the
representative member "))
                If Desiredbudget < 0 Then Throw New NegativeException("Desired
Budget")
                Popularity = CInt(InputBox("Please enter the popularity of the
representative member (an Integer value between 1 and 10)"))
                years = CInt(InputBox("Please enter the number of years of the
representative member (an Integer value between 1 and 10)"))
                Dim tempRep As New cRep(Name, Reputation, Desiredbudget, Popularity,
years)
                txtDisplay.Text += "===== " +
vbNewLine + "Representative " + tempRep.Display
                Members(cMember.GetMemberCount) = tempRep
            Case 2

```

```

        Dim Name As String, Reputation As Integer, Desiredbudget As Double,
Portfolio As Integer
        Name = InputBox("Please enter the name of the minister member ")
        Reputation = CInt(InputBox("Please enter the reputation level of the
minister member (an Integer value between 1 and 10)"))
        Desiredbudget = CDb1(InputBox("Please enter the desired budget of the
minister member "))
        If Desiredbudget < 0 Then Throw New NegativeException("Desired
Budget")
        Portfolio = CInt(InputBox("Please enter the portfolio of the minister
member (an Integer value between 1 and 10)"))
        Dim tempMin As New cMin(Name, Reputation, Desiredbudget, Portfolio)
        txtDisplay.Text += "=====" +
vbNewLine + "Minister " + tempMin.Display
        Members(cMember.GetMemberCount) = tempMin
    End Select
    If cMember.GetMemberCount = NumberOfMembers Then
        MsgBox("All Members Information have been obtained , Dont forget to save
Before Exitinh App", MsgBoxStyle.OkOnly)
    Else
        MsgBox("The Number Of Completed Members is : " +
CStr(cMember.GetMemberCount) + " , click 'Enter Information' Button again to enter Next
Member's information")
    End If
    Catch exneg As NegativeException
        MsgBox("Error Message : " + vbNewLine + exneg.Message)
    Catch ex As Exception
        MsgBox("Error Message : " + vbNewLine + ex.Message)
    End Try
End Sub
#End Region
Private Sub btnAveBudget_Click(sender As Object, e As EventArgs) Handles
btnAveBudget.Click
    'calculates the average of proposed budget
    Try
        Dim total As Double
        For i As Integer = 1 To cMember.GetMemberCount
            total += Members(i).GiveBudgetProposal 'PolyMorphism
        Next
        txtAve.Text = CStr(total / cMember.GetMemberCount) 'Average Calculated And
displayed
    Catch ex As Exception
        MsgBox("Error Message : " + vbNewLine + ex.Message)
    End Try
End Sub

Private Sub btnCreate_Click(sender As Object, e As EventArgs) Handles btnCreate.Click
    'creates the file to save data to
    fs = New FileStream(filename, FileMode.Create, FileAccess.Write)
    fs.Close()
    fs = Nothing
    MsgBox("File has been created")
End Sub

Private Sub btnSave_Click(sender As Object, e As EventArgs) Handles btnSave.Click
    ' saves all data to file
    Try
        fs = New FileStream(filename, FileMode.Open, FileAccess.Write)

```

```

        bf = New BinaryFormatter
        For i As Integer = 1 To cMember.GetMemberCount
            If TypeOf Members(i) Is cMin Then
                bf.Serialize(fs, Members(i))
            End If
        Next
        fs.Close()
        bf = Nothing
        fs = Nothing
        MsgBox("All data has been saved successfully")
    Catch ex As Exception
        MsgBox("Error Message : " + vbNewLine + ex.Message)
        fs.Close()
        bf = Nothing
        fs = Nothing
    End Try
End Sub
Private Sub btnHigh_Click(sender As Object, e As EventArgs) Handles btnHigh.Click
    Try
        fs = New FileStream(filename, FileMode.Open, FileAccess.Read)
        bf = New BinaryFormatter
        Dim name As String, portfolio As Integer
        portfolio = 0
        name = "Doesnt Exist"
        Dim temparrMin(1) As cMin
        Dim numbMin As Integer = 1
        If fs.Length <= 0 Then ' checks if file is empty
            MsgBox("File is empty")
            fs.Close()
            fs = Nothing
            bf = Nothing
            Exit Sub
        End If
        While fs.Position < fs.Length 'reads and converts
            ReDim Preserve temparrMin(numbMin)
            temparrMin(numbMin) = DirectCast(bf.Deserialize(fs), cMin) 'down cast
            numbMin += 1
        End While
        For i As Integer = 1 To temparrMin.Length - 1
            txtDisplay.Text += "===== " + vbNewLine
+ "Minister " + temparrMin(numbMin).Display 'diplays details
            If portfolio < temparrMin(i).Portfolio Then
                portfolio = temparrMin(i).Portfolio
                name = temparrMin(i).Name
            End If
        Next
        txthigh.Text = "Name : " + name + ", Portfolio : " + CStr(portfolio)
'displays highest portfolio
        fs.Close()
        fs = Nothing
        bf = Nothing
    Catch ex As Exception
        fs.Close()
        fs = Nothing
        bf = Nothing
        MsgBox("Error Message : " + vbNewLine + ex.Message)
    End Try
End Sub

```

End Class

```
' *****
' Surname, Initials:Kuni, D
' Student Number:201320596
' Practical: P11
' Class name: NegativeException
' *****

Option Strict On
Option Infer Off
Option Explicit On
Public Class NegativeException
    Inherits Exception
    Public Sub New(cause As String)
        MyBase.New("You Have enter a Negative Value for the input of " + cause)
    End Sub
End Class
```

```
' *****
' Surname, Initials:Kuni, D
' Student Number:201320596
' Practical: P11
' Class name: cMin
' *****

Option Strict On
Option Infer Off
Option Explicit On
<Serializable> Public Class cMin
    Inherits cMember
    Private _portfolio As Integer
#Region "Methods"
    Public Property Portfolio As Integer
        Get
            Return _portfolio
        End Get
        Set(value As Integer)
            _portfolio = checkval(value)
        End Set
    End Property
    Public Overrides Function GiveBudgetProposal() As Double
        Return MyBase.GiveBudgetProposal() * _portfolio
    End Function
    Public Sub New(Name As String, Reputation As Integer, Desiredbudget As Double,
Portfolio As Integer)
        MyBase.New(Name, Reputation, Desiredbudget)
        Me.Portfolio = Portfolio
    End Sub
#End Region
End Class
```

```

' *****
' Surname, Initials:Kuni, D
' Student Number:201320596
' Practical: P11
' Class name: cRep
' *****

Option Strict On
Option Infer Off
Option Explicit On
<Serializable> Public Class cRep
    Inherits cMember
#Region "Attributes"
    Private _Popularity As Integer
    Private _years As Integer
#End Region
#Region "Methods"
    Public Property Popularity As Integer
        Get
            Return _Popularity
        End Get
        Set(value As Integer)
            _Popularity = checkval(value)
        End Set
    End Property
    Public Property Years As Integer
        Get
            Return _years
        End Get
        Set(value As Integer)
            _years = checkval(value)
        End Set
    End Property
    Public Overrides Function GiveBudgetProposal() As Double
        Return _DesiredBudget * (_Popularity / _years)
    End Function
    Public Sub New(Name As String, Reputation As Integer, Desiredbudget As Double,
Popularity As Integer, years As Integer)
        MyBase.New(Name, Reputation, Desiredbudget)
        Me.Popularity = Popularity
        Me.Years = years
    End Sub
#End Region
End Class

```

```

' *****
' Surname, Initials:Kuni, D
' Student Number:201320596
' Practical: P11
' Class name: cMember
' *****

Option Strict On

```

```

Option Infer Off
Option Explicit On
<Serializable> Public Class cMember
#Region "Attributes"
    Protected _Name As String
    Protected _Reputation As Integer
    Private Shared _Count As Integer
    Protected _DesiredBudget As Double
#End Region
#Region "Methods"
    Protected Function checkval(value As Integer) As Integer
        If value < +10 And value >= 1 Then ' checks for valid value
            Return value
        Else
            Return 1 ' default to one
        End If
    End Function
    Public ReadOnly Property Name As String
        Get
            Return _Name
        End Get
    End Property
    Private Function CalculateBudgetModifier() As Double
        Return _Reputation * (1 / 10)
    End Function
    Public Overridable Function GiveBudgetProposal() As Double
        Return CalculateBudgetModifier() * _DesiredBudget
    End Function
    Public Function Display() As String
        Dim i As String = Environment.NewLine
        Display = "Members Name : " + _Name + i
        Display += "Reputation Level : " + CStr(_Reputation) + i
        Display += "Desired Budget : " + CStr(_DesiredBudget) + i
        Display += "Proposed Budget : " + CStr(GiveBudgetProposal()) + i
    End Function
    Public Shared Function GetMemberCount() As Integer
        Return _Count
    End Function
    Public Sub New(Name As String, Reputation As Integer, Desiredbudget As Double)
        _Name = Name
        _Reputation = checkval(Reputation)
        _DesiredBudget = Desiredbudget
        _Count += 1
    End Sub
#End Region
End Class

```