Potential Exam Questions

Prerelease Questions

**Example of a static method**

Public Shared Function NewRandom(ByVal lowerbound As Integer, ByVal upperbound As Integer)

**Example of composite aggregation**

Dim numbers As New NumberMachine

**Example of Inheritance**

Inherits BingoCard

Programming Questions

**Aligning the bingo card**

Public Sub Displaycard() 'prints the card

For x = 0 To 2

For y = 0 To 8

If y > 0 Then

If numbers(x, y) > 9 Then

Console.Write(numbers(x, y) & ",")

Else

Console.Write(" " & numbers(x, y) & ",")

End If

Else

Console.Write(numbers(x, y) & ",")

End If

Next y

Console.WriteLine()

Next x

End Sub

**Loop through all numbers**

In NumberMachine class

Private numberOrder(89) As Integer

Private back As Integer = 89

For x = 1 To 90

numberOrder(x - 1) = x ' sets numberOrder from 1 to 99

Next x

For x = 1 To 1000

num1 = repo.NewRandom(0, 89) ' random number assigned to num1

num2 = repo.NewRandom(0, 89) ' random number assigned to num2

' switches numberOrder(num1) with numberOrder(num2)

temp = numberOrder(num1) ' integer in numberOrder with position of num1 is assigned to temp

numberOrder(num1) = numberOrder(num2) ' integer in numberOrder with position of num2 is assigned to numberOrder(num1)

**Check for corners in the Bingo Bonus game**

Private Function Corners(ByVal calledNumbers As Integer(), ByVal tail As Integer) As Integer

Dim check As Integer

Dim x As Integer

Dim done As Boolean = False

Do

If numbers(0, x) = 0 Then

x += 1

Else

For z = 0 To tail ' 0 to amount of umbers called

If numbers(0, x) = calledNumbers(z) Then

check += 1

End If

Next

done = True

End If

Loop Until done = True

x = 0

done = False

Do

If numbers(2, x) = 0 Then

x += 1

Else

For z = 0 To tail ' 0 to amount of umbers called

If numbers(2, x) = calledNumbers(z) Then

check += 1

End If

done = True

Next

End If

Loop Until done = True

x = 8

done = False

Do

If numbers(0, x) = 0 Then

x -= 1

Else

For z = 0 To tail ' 0 to amount of umbers called

If numbers(0, x) = calledNumbers(z) Then

check += 1

End If

done = True

Next

End If

Loop Until done = True

x = 8

done = False

Do

If numbers(0, x) = 0 Then

x -= 1

Else

For z = 0 To tail ' 0 to amount of umbers called

If numbers(0, x) = calledNumbers(z) Then

check += 1

End If

done = True

Next

End If

Loop Until done = True

Return check

End Function

**Check for horizontal sin Bingo Bonus game**

Private Function Horizontal(ByVal calledNumbers As Integer(), ByVal tail As Integer) As Boolean

Dim matched As Integer

For y = 0 To 2

For x = 0 To 8

If numbers(y, x) <> 0 Then

For z = 0 To tail ' 0 to amount of umbers called

If numbers(y, x) = calledNumbers(z) Then

matched += 1 ' checks if numbers matched with calledNUmbers

End If

Next

End If

Next

If matched = 5 Then

Return True

Else

matched = 0

End If

Next

Return False

End Function

**Allow duck and other words to be called instead of numbers**

Private Function Caller() As Integer

Caller = numbers.nextBall

Console.WriteLine("and the next ball is.....")

If Caller = 11 Then

Console.WriteLine("legs 11")

ElseIf Caller = 2 Then

Console.WriteLine("two little ducks")

Else

Console.WriteLine(Caller)

End If

End Function

**Validate data entry**

Try Catch Everything