

**CS109 Final**  
**100 Points Total**

Name \_\_\_\_\_

This exam is open book and you are welcome to use a computer in the lab. If you are unable to come up with a full answer, write down pseudocode or your thought processes for partial credit. You have 2 hours. Good Luck!

**1. (24 pts) Short Answer.** Provide brief (1-3 sentence) answers to the following:

- a) Describe what difference there is, if any, between the following two subroutines (aside from the words ByVal and ByRef):

```
Public Sub ChangeArray1(ByVal ary() As Integer)
    ary(0) = 5
End Sub
```

```
Public Sub ChangeArray2(ByRef ary() As Integer)
    ary(0) = 5
End Sub
```

- b) How many array entries are created by the following statement?

```
Dim a(10) As Integer
```

- c) Describe what a breakpoint is and how the debugging tool in Visual Studio can be used to track down bugs.

- d) What control would you use if you wanted to execute some code every 5 seconds?

e) How can the OOP (Object Oriented Programming) principle of **inheritance** help us eliminate redundant code?

f) What role does SQL play in the creation of a database application?

## 2. (16 pts) Find the Bugs

All of the following code snippets have one or more bugs. **Identify each one and fix the bug(s)**. Assume the necessary code is in place to make a working program (e.g. the form exists, the code is within some event, etc.).

a) The following should continue until the user enters a number from 1-3:

```
Dim i As Integer = -1
While ((i <> 1) Or (i <> 2) Or (i <> 3))
    i = CInt(InputBox("Enter either 1, 2, or 3"))
End While
```

b) The following code should print “Big” or “Bigger”

```
Dim i As Integer
i = CInt(InputBox("Enter a integer."))
If (0 < i < 10) Then
    Console.WriteLine("Big")
ElseIf (10 <= i) Then
    Console.WriteLine("Bigger")
End If
```

(problem continues on the next page)

- c) The code below should swap the values of X and Y, outputting “2 1”

```
Dim x, y As Integer
x = 1
y = 2
Swap(x, y)
Console.WriteLine(x & " " & y)      ' should output 2 1
```

The swap subroutine is defined as follows:

```
Public Sub Swap(ByVal x As Integer, ByVal y As Integer)
    x = y
    y = x
End Sub
```

- d) There is an error in the classes below that prevents the program from compiling. Find and fix the bug.

```
Public Class SuperHero
    Private name As String
    Public Sub SetName(ByVal newName As String)
        name = newName
    End Sub
End Class

Public Class Superman
    Inherits SuperHero

    Private weakness As String = "Kryptonite"

    Public Sub GetInfo()
        Console.WriteLine("Name: " & name & " Weakness: " _
            & weakness)
    End Sub
End Class
```

### 3. (15 points) Loopy with Arrays

The function below takes as input an array of strings. The array contains a list of names. Complete the function such that it returns **True** if one or more of the entries in the array is the name "Bobo". If "Bobo" is not any of the names in the array then the function should return **False**.

You can use `aryNames.Length` to get the length of the array.  
Assume that data is stored beginning at `aryNames(0)`.

```
Function ContainsBobo(ByVal aryNames() As String) As Boolean
```

```
End Function
```

### 4. (15 pts) Nested Loops

Rewrite the following code such that the functionality remains the same, but uses two while loops instead of the for loops.

```
Dim i, j As Integer
For i = 10 To 0 Step -2
    For j = 1 To i
        Console.Write("*")
    Next
    Console.WriteLine()
Next
```

## 5) (10 points) Classes and Constructors

Given the following class:

```
Public Class MyClass
    Public val As Integer = 1

    Public Sub New()
        val = 2
    End Sub

    Public Sub New(ByVal newNum As Integer)
        val = newNum
    End Sub

    Public Sub Print()
        Console.WriteLine(val)
    End Sub
End Class
```

What would this code output to the Console?

```
Dim r1 As New MyClass
Dim r2 As New MyClass(3)

r1.Print()
r2.Print()
```

## 6. (10 points) Coding Style

a) Listed below are two programs that estimate the square root of a positive number using a technique known as the Babylonian method. For purposes of this problem, it is not necessary to understand the logic behind how the programs work, so don't waste time tracing through all the code logic. Both programs will work, but one uses better coding techniques than the other. **Indicate which one uses superior coding style and why.**

```
Public Class Form1                                     ' Program 1
    Inherits System.Windows.Forms.Form

    Function Average(ByVal num1 As Double, ByVal num2 As Double) As Double
        Return (num1 + num2) / 2
    End Function

    Function SquareRoot(ByVal n As Double) As Double
        Dim i As Integer
        Dim estimate As Double = 1
        Dim q As Double

        For i = 1 To 10
            q = n / estimate
            estimate = Average(q, estimate)
        Next
        Return estimate
    End Function

    Private Sub btnCompute_Click(...) Handles btnCompute.Click
        Dim num As Double
        num = CDb1(TextBox("Enter a positive number."))
        MessageBox.Show("The square root estimate is " & SquareRoot(num))
    End Sub
End Class
```

```
Public Class Form2                                     ' Program 2
    Inherits System.Windows.Forms.Form

    Public q, estimate, num As Double
    Public i As Integer

    Function Average() As Double
        Return (q + estimate) / 2
    End Function

    Sub SquareRoot()
        estimate = 1
        For i = 1 To 10
            q = num / estimate
            estimate = Average()
        Next
    End Sub

    Private Sub btnCompute_Click(...) Handles btnCompute.Click
        num = CDb1(TextBox("Enter a positive number."))
        SquareRoot()
        MessageBox.Show("The square root estimate is " & estimate)
    End Sub
End Class
```

b) Consider the class below:

```
Public Class SimpleClass
    Public val As Integer
End Class
```

We said that it is generally better to make the class variables Private, with a Public property as follows:

```
Public Class BetterSimpleClass
    Private val As Integer

    Public Property Value() As Integer
        Get
            Return val
        End Get
        Set(ByVal Value As Integer)
            val = Value
        End Set
    End Property

End Class
```

Why is the organization of BetterSimpleClass usually considered a better technique than that of SimpleClass?

## 7. (10 points) Food Sales

Now that VB.NET is over and summer is here, it means selling food from your booth at the State Fair. You have an array of foods that you sell, e.g.:

```
Dim aryFoods() As String = {"Elephant Ears", _  
    "Turkey Leg", "Salmon Taco", "Deep Fried Twinkie"}
```

Each food is given an ID, which is its index in the array. For example, 0 refers to Elephant Ears, 1 refers to Turkey Leg, 2 refers to Salmon Taco, and 3 refers to Deep Fried Twinkies.

At the end of the day you get a log of food sales with the ID of each food in the order it is sold. The log is stored in an array. For example, if you sold, in order: two elephant ears, two turkey legs, one elephant ear, one twinkie, one salmon taco, and one twinkie, then the array would look like this:

```
Dim arySaleLog() As Integer = {0, 0, 1, 1, 0, 3, 2, 3}
```

Write a subroutine called `CountSales` that inputs both arrays as parameters and outputs to the Console window the number of sales for each food.

Given the arrays defined above,

```
CountSales(aryFoods, arySaleLog)
```

Would output:

```
You sold 3 Elephant Ears  
You sold 2 Turkey Leg  
You sold 1 Salmon Taco  
You sold 2 Deep Fried Twinkie
```

Make sure that your subroutine works for any array of foods, not just the example given here. For example, it should also work if there were 10 foods in the array, with ID's from 0 to 9. Hint: Create a separate array to count the number of each food item. Use `aryFoods.Length` to get the number of items in the array.



Blank page in case you need extra space for any problem.