**Basic Programming Practice Final Exam with Answers**

**True/False**

**The letter F indicates False answers**

1. F A Select Case statement must have at least 3 cases.
2. F Module-level variables have wider scope than global variables.
3. F Global variables should be used as much as possible.
4. F List variables (also known as arrays) always store exactly 10 elements.
5. The statement **Dim intGrades(1 To 10) As Integer** reserves memory for exactly 10 elements.
6. The statement **intCount = intCount + 1** is a counter statement.
7. The Open statement is used to open random access files.
8. Visual Basic projects can have more than one form.
9. F A file must be closed or a run-time error will occur.
10. F Once you have initially placed a command button on a form, you cannot reposition it in any way even during Design time.
11. F The Len function is used to determine the length of a list variable (array).
12. F You must include the Option Explicit statement in the Form_Load event procedure.
13. A procedure can have 2 or more parameters (also known as arguments).
14. Before you read from or write to a file, you must open it.
15. F Sequential access files are organized as records.
16. The Visible property can be used with menu command objects.
17. F It is wise to leave a menu command enabled at all times.
18. Select Case statements are more similar to If/ElseIf statements than they are to loops.
19. F The Visible property is usually used with message boxes to make them disappear.
20. Another name for an array is a list.
21. F The statement **Dim ArrayExample(15) As String** declares an array of integers.
22. An enabled menu item can be clicked to generate a Click event.
23. Subscript is the term for the position of an element within an array.
24. A listbox control has a Clear method.

For the following set of questions indicate True or False depending on whether syntax errors or runtime errors are present. Do not consider logic errors. Remember that a syntax error is an error that prevents VB from executing a statement or line of code. You can assume that variables or objects with a VB prefix are of the specified data type or control and that they were declared or created properly.

25. There are no syntax errors in the statement: Dim intStudentList(-3 To 5) As Integer
26. There are no syntax errors in the statement: intDiceRoll = Int( Rnd * 6) + 1
27. There are no syntax errors in the statement: Form1.Print "Hello"
28. F There are no syntax errors in the statement: Case Select intChoice
29. There are no syntax errors in the statement: Dim intFirstArray(24) As Integer
30. There are no syntax errors in the statement: Const MAX As Integer = 500
31. There are no syntax errors in the statement: Put #1, 5, strUserName
32. There are no syntax errors in the statement: Close
33. F There are no syntax errors in the statement: txtUserInput.Hide = True
34. F There are no syntax errors in the statement: mnuSave.Enabled False
35. There are no syntax errors in the statement: MsgBox "Click to Continue"
36. There are no syntax errors in the statement: intA(m) = intSum + 1

There are no syntax errors in the statement: For Counter = 0 To 4 Step 0.5

**Determine the Output** – Write your answers on the answer sheet that has been provided.

Deskcheck the following code segments and answer the following questions. If a final answer is a string, make sure that you surround the answer with double quotes ("John_Doe"). Use the underscore character ( _ ) to indicated any blank spaces within string literals. The code segments should not contain any syntax errors. If you believe a syntax error exists, explain it as carefully and thoroughly as possible. Assume that all variables and objects referred to in each code segment have been declared or created with any default values normally supplied by Visual Basic.

```
Dim j As Integer
Dim intMembers(1 To 13) As Integer

For j = 1 To 11
   intMembers(j) = 5
Next j

intMembers(10) = j - 1
```

37. What is the final value of the variable j after the code segment has fully executed? 12
38. How many iterations of the For/Next loop occurred? 11
39. What is the final value of intMembers(10)? 11
40. What is the final value of intMembers(12)? 0

```
Dim intLoop As Integer
Dim strAnswer As String
Dim strOther As String
strAnswer = ""

For intLoop = 4 to 2 Step -1
   strAnswer = strAnswer & "M"
Next k

strOther = Right(strAnswer, 3)
```

41. What is the final value of strAnswer after the code segment has fully executed? "MMM"
42. What is the final value of strOther? "MMM"

```
Dim intList(10) As Integer
Dim J As Integer

Do While (J < 5)
  intList(J) = J * 2
  J = J + 1
Loop

intList(7) = 3 + intList(3)
intList(8) = intList(3 + 2)
intList(1) = intList(9)
```

43. What is the final value of intList(1)? 0
44. What is the final value of intList(2)? 4
45. What is the final value of intList(3)? 6
46. What is the final value of intList(4)? 8
47. What is the final value of intList(5)? 0
48. What is the final value of intList(6)? 0
49. What is the final value of intList(7)? 9
50. What is the final value of intList(8)? 0
51. What is the final value of intList(9)? 0

**Write a Statement or Code Segment** – Write efficient and syntactically correct Visual Basic statements or code segments that fulfill the following tasks. Declare all necessary variables. You must use the variable and object names given in each problem. If you need to use additional variables or objects, use the correct prefixes.

52. Write a statement that declares an array named **intGames** which can store exactly 50 integers. The first element of the array should have a subscript of 12.

    Dim intGames(12 To 61) As Integer

53. Write a statement that opens a file whose path and filename is "C:\questions.dat". Open it as a random access file with read access. The length of each record of the file is 85 bytes and the file number should be #2.

    Open "C:\questions.dat" For Random As #2 Len = 85

54. Write a statement that opens a file whose pathname is "C:\Temp\scores.txt". Open it as a sequential access for output with the file number #1.

    Open "C:\Temp\scores.txt" For Output As #1

55. Write a statement that places a checkmark next to a menu command object named **mnuEditRed**.

    mnuEditRed.Checked = True

56. Write a statement that draws a line segment from the top left corner of a form named **frmMain** to the point 200, 300.

    frmMain.Line (0,0) - (200,300)  Or  frmMain.Line (frmMain.Left, frmMain.Top) - (200, 300)

57. Write a statement that declares a module-level array of 10 **QuestionAndAnswer** user-defined types. Name the array **mudtJeopardyQuestions**. The subscript positions of the array should be 1 through 10.

    Dim mudtJeopardyQuestions(1 To 10) As QuestionAndAnswer

58. Write a statement that assigns the string "blue" to a member variable **strAnswer** of the first element of the array declared in the previous exercise.

    mudtJeopardyQuestions(1).strAnswer = "blue"

59. Write a statement that uses the Rnd function to compute and store a random integer between or including 1 and 6 in the variable **intDiceRoll**.

    intDiceRoll = Int(Rnd * 6) + 1

60. Write a statement that uses the Rnd function to compute and store a random integer between or including 1 and 100 in the variable **intRoulette**.

    intDiceRoll = Int(Rnd * 100) + 1

61. Write a statement that uses the Rnd function to compute and store a random integer between or including 11 and 16 in the variable **intDiceRoll**.

    intDiceRoll = Int(Rnd * 6) + 11

62. Write a code segment that uses a Do/While loop to search an array of integers named **intList** to determine if the value 99 is found in any position within intList. The array has 25 elements with the subscript positions 1 to 25. If the value 99 is found, assign the subscript position of the first found occurrence to the variable **intLocation**. Otherwise the value -1 should be assigned to **intLocation**. You may assume that the array intList has been declared and contains a valid integer value in each position. You must declare all variables properly.

    intPosition = 1
    intLocation = -1

```
Do While (intList(intPosition) <> 99 And intPosition < 26)
   intPosition = intPosition + 1
Loop

If (intPosition = 26) Then
  intLocation = -1
Else
  intLocation = intPosition
End If
```

There are other acceptable solutions to this exercise.

63. Write a full Form_KeyPress event procedure for a form named **frmMain** that causes the PictureBox object **picPlane** to move in the following manner. It should move 10 twips to the left if the lowercase letter 'a' is pressed. It should move 10 twips to the right if the lowercase letter 'b' is pressed.

```
Private Sub Form_KeyPress(KeyAscii As Integer)

  If (KeyAscii = 97) Then
    picPlane.Left = picPlane.Left - 10
  ElseIf (KeyAscii = 98) Then
    picPlane.Left = picPlane.Left + 10
  End If

End Sub
```

64. Write a Do/While loop that moves a shape object named **shpObject** from right to left across a form named **frmMain** until the left edge of shpObject touches or goes past the left edge of the frmMain. On each loop iteration shpObject should move 10 twips.

```
Do While (shpObject.Left > frmMain.Left)
  shpObject.Left = shpObject.Left - 10
Loop
```

65. Write a code segment that could be used to swap the integer values of variables **intThis** and **intThat**. You may assume that the two variables have been declared and that they both contain integer values. Declare any other necessary variables.

```
Dim intTemp As Integer

intTemp = intThis
intThis = intThat
intThat = intTemp
```

66. Write a code segment that uses a For/Next loop and the InputBox function to store 10 strings entered by the user into an array of strings named **strFirstNames** which has subscripts of 1 through 10. You must declare all variables properly.

```
Dim J As Integer
Dim strFirstName(1 To 10) As String

For J = 1 To 10
  strFirstName(J) = InputBox("Enter a string")
Next J
```

67. Write a Select Case statement that uses a control variable **intGrade** to present a message box with the message "Fail" if intGrade is less than 60. If intGrade is greater than or equal to 60 then the message "Pass" should appear in a message box. You may assume that intGrade has been declared and that it holds a valid integer value.

```
Select Case intGrade
Case Is >= 60
  MsgBox "Pass"
Case Is < 60
  MsgBox "Fail"
```

End Select

68. Write a statement that writes the value of the variable **intHighScore** to record number 5 of a random access file opened as file number 2.

    Put #2, 5, intHighScore

69. Write a Do While loop that reads all of the lines of a sequential access file opened as file #1 and stores each line of the file into a listbox named **lstItems**. Make sure that the loop continues until the exact end of the file has been reached. (Hint: Use the EOF function.) Declare any other necessary variables.

    ```
    Dim strTemp As String

    Do While (Not(EOF(1)))
      Input #1, strTemp
      lstItems.AddItem strTemp
    Loop
    ```

70. Write a For Each loop that will close all forms that are currently open in a project. Declare any necessary variables.

    ```
    Dim frmForm As Form

    For Each frmForm in Forms
      Unload frmForm
    Next
    ```

71. Write code for a whole form named **frmMain** that performs the following tasks. When the user clicks a command button called **cmdEnter**, a shape object named **shpVisible** to appear in the very top left corner of the form. In the form's Load event make sure that shpVisible is made invisible.

    ```
    Private Sub Form_Load( )
      shpVisible.Visible = False
      shpVisible.Left = 0
      shpVisible.Top = 0
    End Sub

    Private Sub cmdEnter_Click( )
      shpVisible.Visible = True
    End Sub
    ```

72. Write a general procedure named **DrawSquare** that draws a square in the center of a form named **frmMain**. The square's center should be in the exact center of the form & its width should be 80. You can assume that the form itself is a perfect square and that its ScaleWidth and ScaleHeight properties are both 100.

    ```
    Private Sub DrawSquare()
      frmMain.Line (10, 10) – (90, 10)
      frmMain.Line (90, 10) – (90, 90)
      frmMain.Line (90, 90) – (10, 90)
      frmMain.Line (10, 90) – (10, 10)
    End Sub
    ```

73. Write a function named **intDiceRoll** that randomly returns a whole number between or including 1 and 6.

    ```
    Private Function intDiceRoll() As Integer
      intDiceRoll = Int(Rnd * 6) + 1
    End Function
    ```