

SOLUTIONS MANUAL

Advanced Visual Basic® 2005

FOURTH EDITION



Kip Irvine
Tony Gaddis



SOLUTIONS MANUAL

Advanced Visual Basic® 2005

FOURTH EDITION



Kip Irvine
Tony Gaddis



Advanced Visual Basic 2005

by Kip Irvine and Tony Gaddis, Addison-Wesley Publishing, 2007

Answers to Chapter Review Questions

Last update: 1/17/2007

Chapter 1

Fill-in-the-Blank

1. business rules
2. classes
3. camelCase
4. inheritance
5. garbage collector
6. descriptive
7. reachable

Multiple Choice

1. b
2. a
3. b.
4. c.
5. d.

True or False

1. False
2. True
3. False
4. False, although it is considered good style
5. False. (an instance of Employee is required)

Short Answer

1. No
2. **Doing so permits the variable to be used after the loop finishes—that is a dangerous practice.**
3. **Type three apostrophes in a row.**
4. **Precede the methods with XML-style comments.**
5. **Three levels**
6. **Pass a structure or object containing the required data.**

7. runtime stack
8. presentation (user interface) tier
9. methods
10. business rules tier
11. Only one copy exists for the class, and the variable is shared by all instances
12. `Window.Color = System.Drawing.Color.Red`
13. Code example


```
Sub MyMethod( ByRef param As String )
End Sub
```
14. It will be reclaimed by the garbage collector, which runs in the background.
15. Enclose one or both classes in different namespaces.

What Do You Think?

1. CInt runs slightly faster, and is more familiar to Visual Basic programmers. The System.Convert functions are supported by the other .NET languages such as C++ and C#.
2. To encapsulate, or hide implementation details. We don't want client classes to stop working if the variable's name or type changes.
3. Relational databases now have object-oriented access libraries such as ADO.NET.
4. Example: switch from a Windows interface to a Web interface.
5. Advantage: Hungarian name expresses the variable's type. Disadvantages: prefix hard to remember, not enough prefixes for the large number of data types, confusion about what to do with user-defined types.

Algorithm Workbench

1. Code example:


```
Class Employee
    Private ID As Integer
    Private Salary As Decimal

End Class
```
2. Code example:


```
Public Property LastName() as String
    Get
        Return m_LastName
    End Get
    Set( ByVal Value as String )
        m_LastName = Value
```

```
End Set  
End Property
```

3. **Code example:**

```
Sub ChangeIt( ByRef obj As Student )  
    obj = New Student("Jones")  
End Sub
```

4. **Code example:**

```
Sub New( ByVal pStrength As Integer, ByVal pIntelligence As Integer  
    )  
    Strength = pStrength  
    Intelligence = pIntelligence  
End Sub
```