**SECTION B: Applications:**

1. Understand the Console application.
2. Explain what a Window application is.
3. Explain what a web application is.
4. Understand Class library and Control library.

***Types of Applications:***

* Console – a command-line application that reads commands built into its machine code (fast reaction), from the console.
* Window application – run a Windows-based operating system using GUI (Graphical User Interface that uses graphics) rather than commands to interact with user.

Cost of creating it is greater than cost of creating console app but ease of use of windows app outweighs development costs.

* Web – 1) website that connects user to server that stores information, 2) helps developers make user-friendly web sites without consuming time on HTML.

DIAGRAM: Web application is connected to web server, which connects to Internet which connects to workstation.

* Class library – set of prewritten codes that developers use to simplify development process.
* Control library – collection of controls created for custom use, such as list, box, menu, text box. Example: creating button and specifying its roles and characteristics.
* Smart device application - .NET developers create applications for pocket PCs, cell phones, and other portable devices.

**SECTION C: Object-Oriented Programming**

* Class and Object
* Constructor and Destructor
* Inheritances and its types
* Polymorphism
* Abstraction and Encapsulation

1. Class is a blueprint for an object.

Module Module1

Class Arithmetical

Public Function Add(ByVal a As Integer, ByVal b As Integer)

Return a + b

End Function

End Class

Sub Main()

Dim objArithmetical As New Arithmetical()

Console.WriteLine(objArithmetical.Add(10, 15))

Console.ReadKey()

End Sub

End Module

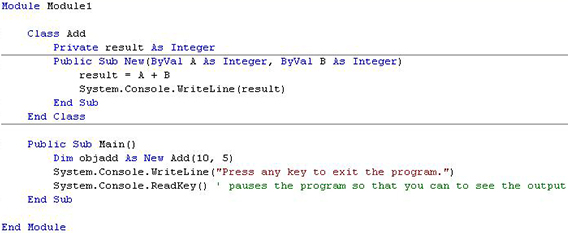
NOTES: Must create an object from the class to directly access Add Function then pass two numbers into Add.

1. Constructor – object’s method to initialize its attributes.

RULES

– cannot return values

* + May or may not have parameters

1. Parameter – takes value of corresponding method/function, and passed to constructor.
2. Destructor cleans up or de-allocates memory when object is destroyed.   
   Finalize and dtor are other names.

Module Module1

Public Class Destructor

Protected Overrides Sub Finalize()

Console.WriteLine("hello")

Console.ReadKey()

Finalize()

End Sub

End Class

Sub Main()

Dim obj As New Destructor ‘ compiler goes into here to get to Class Destructor

End Sub

End Module

***Inheritance:***

* Simple Inheritance: method through which a sub-class is derived from a base class. (single parent)
* Multiple Inheritance: 1) achieved through interface in vb.net 2) sub-class is derived from more than one base class. Example: pocket PC has features of mobile phones and PCs.
* Multilevel Inheritance: multiple generation inheritance. Example: grandfather, father, son.
* Hierarchical Inheritance: multiple subclasses are derived from a single base class. Example: parent derives three children.

**Polymorphism:**

A single unit is shown in different forms, through overloading or overriding.

**Overloading:**

* Overloading means to use members to provide different versions of a property that have same name, but accept different parameters or data types.
* RULES:
  + Function can have same name
  + Data types (parameters) + return types are different
  + If a function has the same parameter and return type as the overloaded one, a counter must note differences between them.

Class Calculator

Public Function Add(ByVal a As Integer, ByVal b As Integer) As Integer

Return a + b

End Function

Public Function Add(ByVal a As Double, ByVal b As Double) As Double

Return a + b

End Function

Public Sub Main()

Dim counter As Calculator

counter = New Calculator()

'pass two integers

Console.WriteLine(counter.Add(1, 5))

'pass two doubles

Console.WriteLine(counter.Add(1, 5))

End Sub

End Class

**Overriding:**

* Overriding means a subclass’s method overrides the functionality of the base class’s existing method. Subclass dog extends Class Animal, and specifies how dogs move.
* Derived classes inherit overridden members.
* <http://www.tutorialspoint.com/java/java_overriding.htm>
* RULES:
  + Method to be overridden must be declared with Overridable keyword

Class Base

'Overridable method uses this keyword

Public Overridable Sub toOverride()

System.Console.WriteLine("Old method")

End Sub

End Class

Class Derived

Inherits Base

Public Shared Sub Main()

Dim overridenResult As New Derived() ‘ Derived has inherited Base features

overridenResult.toOverride()

End Sub

'The sub which overrides the base class method uses this keyword overrides

Public Overrides Sub toOverride()

System.Console.WriteLine("New and improved method")

End Sub

End Class

**Abstraction:**

* Hiding non-essential information and showing just output. Example: code is hidden from a minefield game.
* Hierarchical classifications help user understand complexity of program. Example: car complexity is hidden outside, but contains sub-systems (braking, steering, sound system) which contain more specialized units.
* Abstraction can transform data from process-oriented programs into component objects (concepts) that user can make responsive to user input.

**Encapsulation:**

* Mechanism to bind code and data in a single unit to secure it from access. Class restricts object access.

GOOD REFERENCES:

* Access Modifiers: <http://msdn.microsoft.com/en-us/library/76453kax(v=vs.90).aspx>
* Constructor (dog): <http://www.cis.upenn.edu/~matuszek/General/JavaSyntax/constructors.html>
* Constructor Usage: <http://msdn.microsoft.com/en-us/library/3f80506d(vs.71).aspx>
* Constructors + destructors: <http://www.dotnetheaven.com/article/constructors-and-destructors-in-visual-basic-.net>
* <http://hscripts.com/tutorials/vbnet/constructors-destructors.html>
* Overloading + Overriding: <http://msdn.microsoft.com/en-us/library/zztsbwsx(v=vs.90).aspx>