Abstract Class – 51

* Used only to create derived classes
* Cannot have an object instantiated from the class

Abstraction – 50

* A model of an object, for the purpose of determining the characteristics (properties) and the behaviors(methods) of the object

Accessibility – 76

* Used to determine which classes can use elements
  + Public, private, protected, friend, protected friend
* Program design that allows a disabled person to use your application

Base Class – 51

* The parent class of the current class, from which the current class is inherited

Block Level Scope – 75

* A variable that is visible and accessible only within the block of code in which it is declared

Business Rules – 53

* The logic performed by an application
  + Validation, calculations, updating rules

Business Service Tier – 54

* one segment of a multitier application
  + The class of classes that perform the business rules

Child Class – 51

* An inherited class
  + Aka: Subclass or derived class

Constructor – 56

* A method that automatically executes when an object is instantiated
  + The New method

Data Tier – 54

* One segment of a multitier application
* The class or classes that retrieve and store the data in a database

Derived Class – 51

* An inherited class
  + Aka: subclass or child class

Destructor – 56

* A method that automatically executes when an object is destroyed
  + In VB, the Dispose method is the destructor, which executes at an indeterminate time, whenever the garbage collection occurs

Encapsulation – 50

* The combination of characteristics of an object along with its behaviors

Enum – 81

* The key word used to create a enumeration

Enumeration – 81

* A list of constant values, which must be one of the integer data types

ErrorProvider Component – 64

* Tool for displaying error messages on individual fields
  + Useful for field-level validation

Garbage Collection – 83

* The process in which the .NET Framework destroys unused objects and reclaims memory

Inheritance – 51

* Derive a new class from an existing class
* The new class has all of the public and protected properties and methods of the existing class

Instance Member – 69

* A class variable or property that is not shared by multiple instances of the class
  + Includes instance properties and instance variables

Instance Property – 69

* A class property that is not shared by multiple instances of the class
  + Each instance has its own copy of the property

Instance Variable – 69

* A class variable that is not shared by multiple instances of the class
  + Each instance has its own copy of the variable

Lifetime – 76

* The period of time that a variable remains in scope

Module-Level Scope – 75

* A Private variable that is declared inside any class, structure, or module but outside of any sub procedure or function.
  + Can be used in any procedure of the current class or module

Multitier Application – 53

* An application that separates the functions of an application into multiple classes
  + Separating the user interface from the business logic from the database access

Namespace -74

* A way of organizing classes, interfaces and structures into groups
* Any Public class or identifier in any one namespace must be unique
* To quality the name of an item, include the complete namespace designation
  + System.Web.UI.WebControls

Namespace Scope – 74

* A variable, constant, class, or structure declared with the Public modifier
  + The identifier can be used in any procedure within the namespace, usually the entire application

Overloading – 57

* The two methods have the same name but a different argument list

Overidable – 78

* The keyword used to allow a method in a derived class to override (replace) the method from the superclass

Override – 52

* A method in a derived class that is sued instead of the method in the base class that has the same name
  + An overriding method must have the same argument list as the method in the base class

Parameterized Constructor – 57

* A constructor that requires arguments

Parent Class – 51

* The original class that can be inherited to create a child class
  + Aka: superclass or base class

Polymorphism – 52

* Refers to method names having identical names but different implementations depending on the current object or the arguments supplied

Presentation Tier – 53

* The segment of a multitier application that provides the user interface

Procedure Level Scope – 75

* The scope of any variable that your declare inside a procedure or function, but not within a block
  + The variable can be sued only inside that procedure

Property Procedure – 55

* Public procedures that expose the private properties of a class
  + Contains Get and Set procedures

ReadOnly – 56

* The modifier used in a property procedure to crease a ReadOnly property

Reusability – 51

* The ability to reuse or obtain the functionality from one class of object when you have another similar situation
  + Writing application in components that can be sued in more than one situation

Scope – 74

* The area of the program that can “see” and reference a variable, constant, or method.
  + May be namespace-level, module-level, procedure-level or block-level

Shadow – 78

* Replaces that base-class method in the derived class, but not in any new classes derived from that class
  + A shadowing method need not have the same argument list as the base-class method

Shared Member – 69

* A member of a class that exists once for all instances of the class
  + Includes shared properties and shared variables

Shared Property – 69

* A property of a class that exists once for all instances of the class
  + All instances access the same copy of the property

Shared Variable – 69

* A variable of a class that exists once for all instances of the class
  + Often used to accumulate totals
  + All instances access the same copy of the variable

Subclass – 51

* An inherited class
  + Aka: child class or derived class

SuperClass – 51

* May be inherited to create a subclass
  + Aka: base class or parent class

Throw an exception – 60

* Generate an exception object to indicate that an error has occurred

Throw statement – 61

* Sends or ‘throws’ an exception
  + A Throw statement in a Catch block sends any caught exception up a level

TryParse method – 62

* Allows a value to be tested outside of the Try/Catch proves
  + Uses less system resources

Validating Event – 63

* Event used for testing the contents of a field

Value Keyword – 55

* In a Property Set procedure, used to refer to the incoming value for the property

WriteOnly – 56

* The modifier used in a property procedure to create a property that can be set but not returned