Question Bank –MS .NET

1. What is a computer? (See " Programming Languages.")
2. Name Three Advantages of C# over other programming languages. (See " Enter c#.")
3. What is a comment (to a programmer) & what is it Used For / (See "A Review of the Program.")
4. What does the WriteLine() command do? What is the difference between Write () and WriteLine()? (See " The Fahrenheit to Celsius Conversion Program.")
5. What is the Potential problem using an integer variable when converting a Farhrenheit temperature in Celsius? (See " Demonstrating the Limitations common to all integer types.")
6. What one absolute statements can you make about all C# expressions? (See "Numeric Constants.")
7. Can one type of variable be converted be converted into another? How? (See " Changing Types")
8. What control command is optional on the end of an if Statement? (See "Making Decisions in the World.")
9. From a programmer's perspective, what are the positive and negative effects of indenting embedded clauses? (See "Indenting code for readability.")
10. What is a Boolean? What are the legal values of a bool variable? Name tow operators that are legal for bool variable. (See "Performing Boolean Arithmetic.")
11. Why do int variable name in this book start with an "n"/ (see the Variable naming" sidebar.)
12. Name the Loops that can be used in c# Programming? (See Looping Commands")
13. What are the two special controls that can be used in loops? What are they used for/ (See " Special Controls.")
14. What do we mean by the scope of variable?(See " Scope Rules")
15. What is the most common of all looping constructs? (See "The for loop.")
16. When n starts out as 5, what is the value of n++? (See " why have an increment operator, and why two of them?")
17. Consider the following declaration (See "The Fixed-value array"):
    a. How many elements are in nArray?
    b. What is the index of the first elements in nArray?
    c. What is the index of the last elements in nArray?
18. What is the advantage of the foreach control over the for control when iterating throught arrays? (See " One last Looping Command: for each.")
19. In what way is a class semantically similar to an array? What is the primary difference? (See the first paragraph in " Defining a class.")
20. Define a class pool containing an int element nDepth and a double element dAea. (See Defining Class.")
21. Declare a Pool object and assign it to the reference variable pool1.(See "Creating an Object.")
22. Set the depth of the Pool object to 2 and the surface area to 50. (Don't worry about the units of length or area.)
23. Create a second reference called pool2 that refer to the same Pool object as pool1.
24. Declare a function max ( ) that take no arguments and returns nothing (See " Defining and Using a Function.")
25. Define a function max ( ) that accept two int variable as its arguments.(See "Passing multiple arguments to functions.")
26. Defrine a functions max ( ) that accepts two int varbles as its argument and that returns an int. (See "Returning Values from a Function.")
27. What is a method?
28. using the Pool class the quiz in Session 7, how would a method MaxDepth ( ), which accepts another Pool object and returns the maximum depth, be declared?
29. What would be the full name of the MaxDepth ( ) function? (See " Expanding a Function's Full Name.")
30. What is this ? (See " The this reference")
31. Consider the two method (see " The this reference"):

```
public class MyClass
{
        int nValue;

        public int RetValuel( )
        {
                return nValue
        }

        public int RetValue2( )
```

```
        {
                return this.nvalue;
        }
}
```

Is there any difference between the tow methods?

32. What is the difference between an instance member and a class member?
33. What keyword is used to declare a class member? (See “Class Members”)
34. What keyword is used to declare a class member? (See “ Creating a class member.”)
35. What is the class type of my name?(See “Operating on a String”)
36. What would the call string. Compare (“mystring”, “mystring”) return?(See “Manipulating a string as an object.”)
37. What are the Trim ( ) and Pad Method used for? (See Using the Concatenate methods.”)
38. What are the String. Format ( ) controls used for? (See “String Format Controls.”)
39. What is the purpose of the length property? (See “Review of Simple Arrays”)
40. What is the type of an array of int object? (See the function InitArray ( ) in the “Passing an Array to a Function” section)
41. What is the purpose of the array of strings passed to Main ( )? (See “ Passing an Array to a Function.”)
42. How can a DOS program be executed under windows? (See “ Executing a Command line program from a command prompt window.”)
43. What is a jagged matrix (See The jagged matrix.”)
44. Describe the difference between the way in which the objected programmer and the functional programmer see the world. (See “ Functional nachos.”)
45. Why use the object-oriented programming technique? (See “Why classify Objects in Software? “)
46. Why do I say that Saving Account is a type of BankAccounts? (See “ The IS_A relationship.”)
47. What is the relations between a TV and a picture tube? (See” When to IS_A and When to HAS_A.”)
48. What is the is command for? (See “Avoiding invalid conversion using the is keywords.”)
49. What is polymorphism? (See the introduction to this session)
50. How do you hide a base class member? (See hiding Base class Methods in the Subclass.”)
51. How do you access a hidden member from within the class? (See “Accessing a hidden method from another class.”)
52. How do you access a hidden member form another class? (See “ Accessing a hidden method from another class.”)
53. What is the purpose of the abstract keyword? (See “ The Abstract BankAccount.”)
54. What is the overriding property of an abstract class? (See “The Anstract BankAccount.”)
55. What is a property?
56. What is another name for a static member? Why? (See “Static Properties)
57. What is virtual property? (See “ Virtual properties.”)
58. What is a constructor used for? (See “The C#-Provided Constructor.”)
59. What is the default constructor? (See “The Default Constructor.”)
60. How do your initialize an object with an external Init function? (See “ Initializing an Object Using an External Init Function.”)
61. How do your initialize an object with an internal Init function? (See “ Initializing an Object Using an External Init Function.”)
62. How do you get an object to initialize itself? (See “Initializing an object Directly.”)
63. How can a single class have more than one constructor? How does the program differentiate among constructors? (See “Multiple Constructors.”)
64. Why do such a thing? (See “Multiple Constructors.”)
65. How do you avoid duplication in constructors? (See “Avoiding Duplication Among Constructors.”)
66. Are constructors inherited? (See “Invoking the Default Base Class Constructor.”)
67. How do you pass arguments to the base class constructor? (See “Passing Arguments to the Base Class Constructor.”)
68. What is the destructor? (See “The Destructor”
69. Why divide a program into more than one file? (See the introductory paragraphs.)
70. What is a namespace? (See “ Declaring a namespace.”)
71. What are the five access control keywords? (See “Namespaces and Class Access.”)
72. Why do we need error handling? (See “ Why do you Need a New Error Mechanism?”)

73. What new mechanism does C# introduce for error handling? (See " Demonstrating the exception mechanism.")
74. How can we extent the exception class? (See "Extending the Exception class.")
75. How can we use multiple catch blocks for handling? (See "Multiple catch blocks.")
76. What are the key elements of an exception handler? (See " Multiple catch blocks.")
77. What are cascading exceptions? (See Cascading exceptions.")
78. What is the difference between a reference type and a value type?(See " Declaring a C# struct.")
79. What is meant by *unifying types* and how is boxing involved? (See "unified Type System.")
80. What is the CAN_BE_USED_AS relationship? How does it differ from the IS_A relationship? (See . " The Interface.")
81. What is an interface? (See " Examining an Interface.")
82. What are predefined interfaces? (See "Predefined interfaces.")
83. What is an abstract interface? (See " Abstract Interface.")
84. What is a delegate? (See the introduction to this session.)
85. What is a callback function? (See the " Why bother?" Sidebar.)
86. What is a disadvantage of an array? (See " Some advantage and disadvantage of the array.")
87. Name an alternative collection that does not suffer from the disadvantage of an array (it may carry its own disadvantages).(See the section after" some advantages and disadvantage of the array.")
88. Name a third of collection.(See "Other collection types.")
89. What is an indexer? (See "Indexers")