1. Declare a C# base class that cannot be instantiated directly, but must be overridden to create a class that can be instantiated. (2 points)

2. Alter this program to trap exceptions thrown from the DoSomething() method and log them to the TextWriter instance. (7 points)

class Program

{

private static TextWriter logFile = File.AppendText("log.txt");

static void Main(string[] args)

{

DoSomething();

}

}

3. Which is the correct approach to using an object that implements IDisposable in C#: (10 points)

1.

void DoSomething()

{

var connection = new SqlConnection(ConnectionString);

// Use connection

}

2.

void DoSomething()

{

var connection = new SqlConnection(ConnectionString);

// Use connection

connection.Dispose();

}

3.

void DoSomething()

{

using (var connection = new SqlConnection(ConnectionString))

{

// Use connection

}

}

4. What's wrong with this code? (5 points)

// Walk up the item's parents, printing the name to the console

void DoWalkThePath(Item item)

{

while (item != null)

{

Console.WriteLine(item.Name);

}

}

class Item

{

public string Name;

public Item Parent;

}

5. The following function is supposed to return the largest of the 3 integers passed as parameters. Implement the test method to verify that this function works as expected. (8 points)

int LargestOfThree(int num1, int num2, int num3);

[TestMethod()]

public void TestLargestOfThree()

{

}

6. Given the following C# class definition, use the Enumerable extension methods in LINQ to return the first 3 Names that start with X as a comma-delimited list. (10 points)

class Product

{

public string Name { get; set; }

public double Cost { get; set; }

}

static string GetFirst3XNames(IEnumerable<Product> products)

{

// Enter code here

}

7. Given the following XML file, use the XPath support in .NET to write out all the names, one on each line: (8 points)

<Root>

<Item>

<Name>Smith</Name>

</Item>

<Item>

<Name>Brown</Name>

</Item>

<Item>

<Name>Jones</Name>

</Item>

<Item>

<Name>John</Name>

</Item>

</Root>

8. Implement the following method using reflection. The method should locate a public property on instance with the name passed in via the propertyName parameter, and set the property's value to propertyValue. (10 points)

void SetPropertyOnType<T>(T instance, string propertyName, string propertyValue)

{

}

9. Given the following ASPX file, which method can be used to read the JavaScript random number in the server-side code in the page: (10 points)

<%@ Page Language="C#"%>

<%

// How to read number here?

%>

<html xmlns="http://www.w3.org/1999/xhtml" >

<head runat="server">

<script type="text/javascript">

var number = (Math.random() \* 1000);

</script>

</head>

<body>

<form id="form1" runat="server">

</form>

</body>

</html>

Possible answers:

1. Cookies
2. Form fields
3. JavaScript postback via Ajax or form post.
4. All of the above
5. None of the above, the C# code can read JavaScript variables directly.

10. In the MVC architecture, should a Controller contain business logic? (5 points)

11. In the MVC architecture, should a Controller contain code to load data from the database? (5 points)

12. In the MVC architecture, should a Model contain business logic? (5 points)

13. In the MVC architecture, should a Controller be responsible for executing the business logic based on input parameters? (5 points)

14. In the MVC architecture, do Repository classes contain code to load the models from the database? (5 points)

15. In the MVC architecture, should Services contain code to generate the output shown to users? (5 points)