.NET Engineering Examination (Rev 2)

***Problem Solving***

1. The exception occurred because a null reference type was passed to the .Add method of the instance (b) of class Basket. Since f.SeedCount is a value type and was initialized to zero, then it's not that parameter. I suspect that there was a List<Fruit> that was being manually incremented and passed to Add() (instead of through a foreach loop) and it was incremented past the last item in the list.

2. Line three has "i<= 3". With three items in the array (arrays are zero based), the loop try to go past the end of the array.

Line four says "<1", so first problem notwithstanding, the loop will only give us the first name. That may be what is wanted but the problem is that string p is never initialized to a value, therefore it will give us a null reference exception.

***.Net***

1. What is a process? On a given machine, a process is the code and data for each program running. Each process is isolated in terms of security.

What is an AppDomain? On IIS, the AppDomain is the boundary for code and data for applications. Like processes, the have their own security access.

What are threads? Threads are independent execution paths for code, running under the context of the AppDomain.

2. When to you use reflection? Whenever it is necessary to list types or properties of objects at runtime. When making modifications and in debug mode, it can be useful to know what is available in the assembly for use or what to avoid in terms of potentially hiding a like named object. When operating on generics, it is useful for determining the proper actions to apply to an object based on the runtime type.

***XML***

1. What does DOM stand for and what does it do? Document Object Model. Provides a structured representation of the objects available in the browser. It provides an API with which to handle these objects.

2. What is the difference between XmlNode and XmlElement? XmlNode is the base class for XmlElement. An XmlElement is a representation of specific data as text or other XMLElements. It includes everything from the start tag and end tag of that element.

**a) //Book[@Genre='Horror']**

**b) /Book/[substring-after( Price, '$')] > 10)**

**c) //Book[not(contains(@Genre,'Drama'))]**

3. What is the output of this code?

**Hello World is a test**

**No string second try**

4. Define the following:

**abstract class** - An abstract cannot be implemented, it must be derived from by a new class. An abstract class will contain at least one abstract method which must be implemented in the derived class. The abstract class can also contain non-abstract methods which may be changed in the derived class with the beginning keyword "override".

**.NET primitive type -** A primitive type will be defined in the CLR as having type.IsPrimitive set to true. The distinction between value and reference types is not the same as the difference between primitive and non-primitive types as some value types can be primitive and others not. Some examples of primitives are the int types, bool, char and byte types.

**generic constraint -** A generic type is defined by the "where" keyword in terms of the interfaces from which it derives, whether it is a value or a reference type, whether or not it takes a parameterless constructor (new), and specifying a base class.

***SQL***

1. select \* from customers

2. select distinct C.\* from customers C

inner join Orders O on C.id = O.CUSTOMER\_ID

3. select C.\* from Customers C

where C.id not in (

select O.Customer\_ID from customers C

inner join Orders O on C.id = O.CUSTOMER\_ID)

4. Clustered index on Customers.ID since it joins the orders table in query 3. Clustered index on Orders.ID since it will most likely be used to link future tables to it. Non-clustered index on Orders.Customer\_ID since it is the foreign key that joins the query in 3., above.

5. select C.Name [Name], C.Email [Email], count(O.id) [Total Orders] from customers C

inner join Orders O on C.id = O.CUSTOMER\_ID

group by [Name], [Email]

6. select C.Name [Name], C.Email [Email], count(O.id) [Total Orders] from customers C

inner join Orders O on C.id = O.CUSTOMER\_ID

group by [Name], [Email]

having count(O.id) between 1 and 4

***Research***

1. Very often, I use the **FileWatcher** component of .NET. For some reason, when a file event happens, my handlers get called more than once. Why does this happen, and how can I work around it?

Answer. This happens because, for instance when a file is created, there are more than one processes involved with that creation, each of which calls the handler. This is a well-known problem and there are a number of work arounds for it. Adhering to the KISS approach, the best one seems to be this, where even though the event continues to fire, the delegate is supressed after the first occurance.

private int fireCount = 0;

private void inputFileWatcher\_Changed(object sender, FileSystemEventArgs e)

{

fireCount++;

if (fireCount == 1)

{

MessageBox.Show("Fired only once!!");

dowork();

}

else

{

fireCount = 0;

}

}

}

2. I am using Intersoft WebCombo. I want to use multiple column mode for my combobox, but I don’t know how. How do I do this?

Here are the steps to accomplish that:

* [Bind WebCombo to AccessDataSource](http://www.intersoftsolutions.com/Support/TutorialConverter.aspx?product=WebCombo&url=Binding%20WebCombo%20to%20AccessDataSource%20control.html).
* Open **WebCombo Designer** - **Advanced Settings - LayoutSettings**.
* Set **ComboMode** to **MultipleColumns**.
* Go to **Column and Rows** and select **Displays multiple columns in the result box of this control instance**.
* Add the columns that you want to display.
* Click **OK** to apply the changes and run the project.

3. I get a NullReferenceException in the RadAjaxManager from Telerik when running AddAjaxSetting – any ideas on how to fix?

There is a control parameter in the Add function which represents the control that initiated the update. There must be a value provided for ID.

AjaxSetting

setting =newAjaxSetting();

AjaxUpdatedControltarget =newAjaxUpdatedControl(rgdTLA.ID, rlp.ID );

setting.UpdatedControls.Add(target);

setting.AjaxControlID = lnk.ID ;

radAjaxManager.AjaxSettings.Add(setting);

***Design Patterns***

1. Describe the **Strategy** pattern and when you use it, giving a concrete example

This pattern encapsulates a family of related algorithms into separate classes allowing each to serve a single purpose. An example of this pattern's use can be seen in a shipment calculator for an online ordering application. Whereas there are a number of different shipping providers and each with a separate cost calculation, those calculations would be contained in a single class. In the implementation, the cost, calculated on the same order but delegated to the different provider's classes, would be provided for comparison at the end. Any changes in the calculation algorithm for the providers would be constrained to the associated class, thus upholding the open/closed principle.

1. Describe the **Unit Of Work** pattern, and what problems it solves. This pattern has a lot in common with a transaction as described in TSQL where a series of operations are performed and finally committed to a persistence medium when all are complete. It supports the concept of atomicity. Through the use of this pattern, three problems are solved: 1) Business logic is isolated from the database code, 2) Business logic does not have to track completion of the entire transaction, 3) Marshals multiple individual processes into a meaningful object to be persisted to media.