**Examination #5**

Name:\_\_\_\_\_\_\_\_ **Instructor’s Solution** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ SUID:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This is a closed book examination. Please place all your books on the floor beside you. You may keep one page of notes on your desktop in addition to this exam package. All examinations will be collected promptly at the end of the class period. Please be prepared to quickly hand in your examination at that time.

If you have any questions, please do not leave your seat. Raise your hand and I will come to your desk to discuss your question. I will answer all questions about the meaning of the wording of any question. I may choose not to answer other questions.

You will find it helpful to review all questions before beginning. All questions are given equal weight for grading, but not all questions have the same difficulty. Therefore, it is very much to your advantage to answer first those questions you believe to be easiest.

1. Draw a package diagram for the TestHarness of Project #4.   
     
   Answer:
2. Suppose that the TestHarness, of Project #4, uses a File Service with a service method:  
     
    Stream downLoadFile(string filename);  
     
   Write all the code for creating, starting, and running a thread that uses this method to acquire the files it needs to run a test, assuming that the filenames are stored in a List<string>.  
     
   Answer:

1. What is a Repository, and why is its use an important part of the Software Collaboration Federation you will discuss in Project #5?  
     
   Answer:
2. Write a code fragment to define a lambda and bind to an Action delegate, where the lambda calls a function string convert(int i) { … }. You may assume the convert function is a public member of the class in which the lambda is being defined. You are not required to provide the code for convert, but just use it to define the lambda’s processing. Furthermore, the Action delegate has no generic parameters.   
     
   Answer:

1. What does the term “Software Architecture” mean? What are the most important things to discuss when talking about the architecture of some software system?  
     
   Answer:

1. WCF allows the construction of a proxy for a service that may not yet be running. When the first call is made to the service, using the proxy, it fails with an exception if the service is not available. Write a wrapper class for the proxy that retries the first call a finite number of times with brief waits between calls if the call initially fails.   
     
   Answer:  
     
   Main issue here is that service methods may have different signatures, so must use a lambda:

class ReTry // answer starts here

{

public int numRetries { get; set; } = 10;

public int waitMilliSec { get; set; } = 50;

public void doWithReTry(Action action)

{

for (int i = 0; i < numRetries; ++i)

{

try

{

action.Invoke();

break;

}

catch

{

Thread.Sleep(waitMilliSec);

Console.Write("\n proxy call failed");

Console.Write("\n retrying");

}

}

}

} // answer ends here

public class Proxy

{

static int count = 0;

public void someProxyMethod(string msg)

{

if (++count < 4)

throw new Exception("proxy call failed");

Console.Write("\n proxy call succeeded with message \"{0}\"", msg);

}

}

class MT4Q6

{

static void Main(string[] args)

{

Console.Write("\n MT4Q4 - Retry Proxy");

Console.Write("\n =====================\n");

ReTry reTry = new ReTry();

Proxy proxy = new Proxy();

string msg = "Hello from MT4Q4";

reTry.doWithReTry(() => { proxy.someProxyMethod(msg); });

Console.Write("\n\n");

}

}

1. What is the purpose of the System.Type class? Write a small fragment of code to show one way to use it with an object. Repeat for use with a C# class.  
     
   Answer:  
     
   The Type class is a container for information gathered about an object or class using reflection.  
     
    **Use with object:**

MT3Q3 mt3Q3 = new MT3Q3();

Type t = mt3Q3.GetType();

Console.Write("\n name of this class is {0}", t.Name);

Console.Write("\n namesapce is {0}", t.Namespace);

**Use with class:**

Type t = typeof(MT3Q3);

Console.Write("\n name of this class is {0}", t.Name);

Console.Write("\n namesapce is {0}", t.Namespace);

See MT3Q3.cs for code using these fragments.