**1) What is the advantage of an Enum over (for example) an *int*?**

- The advantage of an ENUM over an int is clarity. It is useful for unorganized categorical data. For example, the question “What day of the week is it?” and you answer “5”, the answer depends on where you are indexing from (Sunday? Monday?) whereas “Friday” from an ENUM is more specific.

**2) When might you prefer a struct over a class?**

-With classes, it is possible for two variables to reference the same object, which makes it possible that any operation performed on one of those variables would affect the other. With Structs, the variables each have their own copy of the data and it is not possible for operations on one to affect the other.

**3) What happens if you create a struct without using the *new* keyword?**

-If you create a struct without using the *new* keyword, it does not call any constructors, so all the members remain unsigned.

**4) What is the purpose of CIL?**

-The CIL stands for Common Intermediate Language. CIL is CPU and platform-independent instructions that can be executed in any environment supporting the Common Language Infrastructure, such as the .NET run time or the cross-platform Mono run time. These instructions are later processed by run time compiler and is then converted into native language. CIL is an object-oriented assembly language and is entirely stack-based.

**5) What does the CLR do?**

The CLR converts the CIL into instructions that can be understood and executed by the processor on your computer.

**6) What alternative to native code does the Universal Windows Platform provide for unmanaged applications?**

-The CLR converts the CIL into instructions that can be understood and executed by the processor on your computer.

**Apply**

*Classes and Objects*

7) Define a class with:

-One public field

-One private field

-A default constructor

-A 2-argument constructor which sets your fields

-A deconstructor

-SEE BELOW #8

8) Instantiate an instance of your class using the 2-argument constructor. Deconstruct the instantiated object into a tuple.

namespace Homework

{

class Program

{

static void Main(string[] args)

{

Instrument instrument = new Instrument(); //instantiate new class

}

}

public class Instrument

{

public string Name;

private string Category;

public Instrument() : this("DefaultName", "DefaultCategory")

{

}

public Instrument(string name, string category)

{

this.Name = name;

this.Category = category;

}

public void Deconstruct(out string name, out string category)

{

name = this.Name;

category = this.Category;

}

}

}

\*I don’t know if I did this correctly or not. Honestly at this point I’m getting confused.