CS371 Quiz 2 Thurs Jan 12th, 2012

Name: \_\_\_\_Gabe\_\_\_

Please note that this is a closed book quiz. Complete this document and save it in [\\CS1\CS\_Students\your name\CS371\Quiz\Quiz2.docx](file:///\\CS1\CS_Students\your%20name\CS371\Quiz\Quiz2.docx)

1. (2 points) You are given the follow defintion:

class MyClass {

public static void myMethod() {

…

}

}

Show the most efficient way to invoke the method **myMethod()** in your program.

using(Myclass temp = new Mysclass())

{

temp.myMethod();

}

**[comment: -1 no need to create an object.**

**MyClass.myMethod();**

**]**

1. (4 points) Implement the method **LetsGo()** in the sub-classes **Muppet**, **Elmo**, and **BigBird** to ensure polymorphic behavior is implemented correctly. **LetsGo()** just prints some text (you choose) to the console.

Public interface ITraveller {

void LetsGo();

}

Class Muppet : ITraveller

{

public virtual void LetsGo() { Console.WriteLine(“This worked in Muppet”);

}

class Elmo : Muppet

{

public override void LetsGo() { Console.WriteLine(“This worked in Elmo”);

}

Class BigBird : Muppet

{

public override void LetsGo() { Console.WriteLine(“This worked in BigBird”);

}

**[good!]**

1. (2 points) Given the array of **Muppet** objects defined below (assume that it has been initialized properly), write a **foreach** loop that only calls the **LetsGo()** method for **Elmo** objects.

**Muppet [] SesameSt = new Muppet[100];**

**// SesameSt is initialized correctly here …**

foreach(Muppet obj in SesameSt)

{  
 if(obj == Elmo)

((Elmo)obj).LetsGo();

}

**[comment: -1**

**If (obj is Elmo)**

**obj.LetsGo(); // no cast needed**

**]**

1. (2 points) **Define** a delegate type and use it to **declare** a delegate called **CheckMethod** to hold a reference to the method defined below:

bool withInValidRange(decimal x) {

return x > 3M && x < 100M;

}

delegate bool CheckMethod(decimal y){

return withInValidRange(y);

}

**[comment: -1**

**delegate bool delname(decimal x);**

**delname CheckMethod = withInValidRange;**

**]**

1. (2 points) Write a lambda expression to represent the same method logic in Q4.

CheckMethod(someDecimal => if(x > 3M && x < 100M) ? true : false);

**[comment: -1**

**(x) => {return x > 3M && x < 100M;}**

**]**

1. (4 points) Define an exception class called **MyGreatException**.

Class MyGreatException : Exception

{  
 public MyGreatException() {}

public MyGreatException(string message) : base(message) {}

public MyGreatException(string message, Exception inner) : base(message, inner) {}

}

**True/False** Questions (each question is worth 2 points)

\_\_F\_\_ An abstract class can only contain abstract methods and properties.

\_\_T\_\_ A C# class can inherit more than one **interface** type.

\_\_T\_\_ The **ToString()** method is a virtual method defined in the Object class.

\_\_T\_\_ An **extension method** must be defined as a **static method** in a **top-level static class**.

\_\_T\_\_ A class must implement the **IComparable** interface if objects of that class needs to be sorted.

\_\_T\_\_ I like C# better than C++.