```csharp
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace test4First
{
    //Q1
    class student
    {
        private int admno;
        private string sname;
        private float eng, math, science;
        private float total;

        private float ctotal()
        {
            float ans = eng + math + science;
            return ans;
        }

        public void Takedata(int ad, string name, float e, float m, float s)
        {
            admno = ad;
            sname = name;
            eng = e;
            math = m;
            science = s;
            total = ctotal();
        }

        public void Showdata()
        {
            Console.WriteLine("Admno is {0}", admno);
            Console.WriteLine("Name is {0}", sname);
            Console.WriteLine("Eng is {0}, math is {1}, science is {2}", eng, math,
science);
            Console.WriteLine("Total is {0}", total);
        }
    }

    //Q2
```

```csharp
class batsman
{
    private int [] bcode;
    private string bname;
    private int innings, notout, runs;
    private float batavg;

    private float calcavg()
    {
        float ans = runs / (innings - notout);
        return ans;
    }

    public void readdata(int[] code, string name, int input, int noout, int run)
    {
        bcode = code;
        bname = name;
        innings = input;
        notout = noout;
        runs = run;
        batavg = calcavg();
    }

    public void displaydata()
    {
        Console.WriteLine("Bcode is {0}", bcode);
        Console.WriteLine("Name is {0}", bname);
        Console.WriteLine("Innings is {0}", innings);
        Console.WriteLine("Runs is {0}", runs);
        Console.WriteLine("Batavg is {0}", batavg);
    }
}

//Q3
class Q3
{
    public static void Main (string[] args)
    {
        linePrinter("story.txt");
    }

    static void linePrinter(string filename)
    {
        StreamReader reader = new StreamReader(@filename);
```

```csharp
            int count=0;
            string buffer;
            using(reader)
            {
                do
                {
                    buffer = reader.ReadLine();
                    if (buffer[0] != 'A')
                        count++;
                } while(!reader.EndOfStream);

                Console.WriteLine("Output: "+ count);
            }
        }
    }

//Q4
class Animal
{
    public void eat()
    {
        Console.WriteLine("The animal is eating!");
    }

    public void breath()
    {
        Console.WriteLine("The animal is breathing!");
    }

    public void sleep()
    {
        Console.WriteLine("The animal is sleeping!");
    }
}

class Turtle: Animal
{
    public void retract()
    {
        Console.WriteLine("The turtle is retracting!");
    }
}

class Dog: Animal
```

```csharp
    {
        public void bark()
        {
            Console.WriteLine("The dog is barking!");
        }
    }

    class Bird: Animal
    {
        public void fly()
        {
            Console.WriteLine("The animal is flying!");
        }
    }
}

namespace test4Second
{
    //Q5
    public class Q5
    {
        public static IEnumerable<int> myFilter(IEnumerable<int> input)
        {
            IEnumerable<int> ret;
            ret = input.Where(x => x % 2 != 0).Where(x => x % 5 == 0);
            return ret;
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            int[] numbers1 = { 1,3,5,6,7,8,9,10,11,12,13,29,30,31,32,33 };
            int[]                          numbers2                          =
{ 30,31,40,41,50,51,60,61,70,71,72,74,75,100,101,101,200 };
            int[] arr= Enumerable.Concat(numbers1, numbers2).ToArray();
            var answer = Q5.myFilter(arr);
            foreach( int i in answer )
            {
                Console.WriteLine(i);
            }
        }
    }
```

```csharp
//Q6
public class Q6
{
    public static IEnumerable<int> myFilter(IEnumerable<int> input)
    {
        IEnumerable<int> ret;
        ret = input.Select(x => x * x).Where(x => x % 2 != 0);
        return ret;
    }
}

class Program
{
    static void Main(string[] args)
    {
        int[] numbers3 = { -5, -6, -7, -8, 47, 50, 60, 1000, 2000, 3000 };
        var answer = Q6.myFilter(numbers3);
        foreach( int i in answer )
        {
            Console.WriteLine(i);
        }
    }
}

//Q7
public class Q7
{
    public static IEnumerable<int> myFilter(IEnumerable<int> input)
    {
        IEnumerable<int> ret;
        ret = input.Where(x => isPrime(x));
        return ret;
    }

    public static bool isPrime(int x)
    {
        int a = 0;
        for (int i = 1; i <= x; i++)
            if (x % i == 0)
                    a++;
        if (a == 2)
            return true;
        else
```

```
                return false;
            }
        }

        class Program
        {
            static void Main(string[] args)
            {
                int[]                    numbers2                    =
{ 30,31,40,41,50,51,60,61,70,71,72,74,75,100,101,101,200 };
                var answer = Q7.myFilter(numbers2);
                foreach( int i in answer )
                {
                    Console.WriteLine(i);
                }
            }
        }
}
```

Q8

C# is an object-oriented language that supports component-oriented programming. C#
provides properties, methods, and events for the programming model, including
features that provide declarative information about components. This means that C#
has good applicability in game development. In fact, one of my game projects this
semester was completed on Unity by using C#. Using C# can more conveniently
operate the prefab required by the game and add features to it.