```
// Introduction to Windows and Web Applications in C#
// demonstration student/address/state from class 05
// copyright 2013 Bruce M Reynolds

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO;
using System.Diagnostics;

namespace Week05Demo
{
    class Program
    {
        static void Main(string[] args)
        {
            BasicDemo();
            DemoIO();
            DemoIOWithException();
        }

        static void BasicDemo()
        {
            Student student = new Student("Ada");
            student.Address = new Address("Seattle", State.Washington);

            // Notice that we can set the student's grades, even though the
            // Grades property is read only (the Grades property includes
            // only a "get" and not a "set").
            student.Grades[0] = 80;
            student.Grades[1] = 100;
            student.Grades[2] = 95;

            // Because the Grades property is read only, we cannot do the following,
            // which sets the student's grades array to a new array:
            // student.Grades = new int[] { 80, 100, 95 };

            // this calls our student's ToString method
            Console.WriteLine(student);

            // demonstrate how to output diagnostics to the output window.
            // Make sure you have "using System.Diagnostics" at the top of the file.
            Debug.WriteLine(Student.StudentTestCode());

            // call each version of our overloaded function
            Console.WriteLine("Average of all grades is {0}", student.CalculateAverage());
```

```
            Console.WriteLine("Average of two grades is {0}", student.CalculateAverage(2));
        }

        // use FileStream and StreamReader & StreamWriter to save/load a student
        static void DemoIO()
        {
            Student student = new Student("Barney", new Address("Bellevue", State.Washington));

            // write out a student's details to a save file
            string fileName = "savefile.txt";
            FileStream output = new FileStream(fileName, FileMode.Create, FileAccess.Write);
            StreamWriter writer = new StreamWriter(output);
            student.WriteToFile(writer);
            writer.Close();

            // read in a student from a save file
            FileStream input = new FileStream(fileName, FileMode.Open, FileAccess.Read);
            StreamReader reader = new StreamReader(input);
            Student copy = Student.ReadFromFile(reader);
            reader.Close();
        }

        // use FileStream and StreamReader & StreamWriter to save/load a student
        static void DemoIOWithException()
        {
            string fileName = "savefile.txt";
            try
            {
                // read in a student from a save file
                FileStream input = new FileStream(fileName, FileMode.Open, FileAccess.Read);
                StreamReader reader = new StreamReader(input);
                Student copy = Student.ReadFromFile(reader);
                reader.Close();
            }
            catch (IOException e)
            {
                Debug.WriteLine("IO errror: " + e.Message);
            }
            catch (FormatException e)
            {
                Debug.WriteLine("Format errror: " + e.Message);
            }
        }
    }
}
```