```
// Introduction to Windows and Web Applications in C#
// demonstration student/address/state from class 5
// copyright 2013 Bruce M Reynolds

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

namespace Week05Demo
{
    class Student
    {
        public Address Address;

        private string name = "";
        public string Name
        {
            get { return name; }
            set { name = value; }
        }

        private int[] grades = new int[3];
        public int[] Grades
        {
            get
            {
                return grades;
            }
        }

        // one of two constructors
        public Student(string name, Address address)
        {
            Name = name;
            Address = address;
        }

        // one of two constructors
        public Student(string name)
        {
            Name = name;
        }

        // override the default System.Object.ToString()
        // to display the details of this student
        public override string ToString()
        {
            string output = Name + "\n" + Address.ToString();
            for (int week = 0; week < grades.Length; week++)
            {
                output += string.Format(
                    "Week {0}: {1}\n", week + 1, grades[week]);
            }
            return output;
        }
```

```
// average all the grades
public int CalculateAverage()
{
    return CalculateAverage(grades.Length);
}

// average the specified number of grades
public int CalculateAverage(int howManyWeeks)
{
    int sum = 0;
    for (int i = 0; i < howManyWeeks; i++)
    {
        sum += grades[i];
    }
    int average = sum / howManyWeeks;
    return average;
}

// save the student to a file
public void WriteToFile(StreamWriter writer)
{
    writer.WriteLine(Name);
    Address.WriteToFile(writer);
    foreach (int grade in Grades)
    {
        writer.WriteLine(grade);
    }
}

// load a student from a file. Since there is no
// existing student until the student's details are
// loaded from the file, this method is "static" and
// constructs and returns the newly created student.
public static Student ReadFromFile(StreamReader reader)
{
    string name = reader.ReadLine();
    Address address = Address.ReadFromFile(reader);

    Student student = new Student(name, address);

    for (int index = 0; index < 3; index++)
    {
        string grade = reader.ReadLine();
        student.Grades[index] = Int32.Parse(grade);
    }
    return student;
}
```

```
        // create three students and return a string with
        // the three students' ToString() results
        // concatenated together.
        public static string StudentTestCode()
        {
            Student Grace = new Student("Grace Hopper",
                new Address("Portland", State.Oregon));
            Student Ada = new Student("Ada Lovelace",
                new Address("Seattle", State.Washington));
            Student Roberta = new Student("Roberta Williams",
                new Address("Missoula", State.Montana));

            string output = "";
            Student[] students = { Grace, Ada, Roberta };
            foreach (Student student in students)
            {
                output += student.ToString();
            }
            return output;
        }
    }
}
```