**SD2 Practical**

**Relations – Unit 5**

1. Start Visual Studio and open the solution Relations1 (Relations1.sln) which will have been downloaded along with this document.
2. Examine the project Relations1, contains two classes Person and House. There is a M:1 relation required between House and Person. At present this is implemented using an Array. Can you see the major problem that exists in the current implementation?
3. Rewrite Relations1 to use a List object in place of the array.

HINTS:

1. You will need to declare the list as List<Person>
2. To loop through items in the list use a foreach loop

foreach (Person p in people)...

1. More info on the List class may be found at <http://msdn.microsoft.com/en-us/library/6sh2ey19.aspx>. Use the menu on the left to access info regarding properties and methods
2. Rewrite Relations1 to use a List object in place of the array.
3. Create a new WPF project and call it UniRecords.

A University has many students, each student has the attributes name, address, date of birth and matriculation number. Students are enrole within the University.

Create a Student class with appropriate properties.

Create a University class, which should have a 1:M relationship with Student.

Create an instance of a University class within your main window (the one created by default when you created the WPF project). Create two buttons, one called Add Student and the other Show students

* 1. Create an event handler for Add Student that creates a number of student objects like this:

Student peter = new Student();

peter.name = "Peter";

peter.matric = "1234/1";

peter.course = "Computing";

Make sure that you add your student objects to the list.

* 1. Create a handler for the Show Students button that uses a foreach loop to access each student object and show its form. Use the Students’ .Show() method to display the form.

1. Now modify your University window so that it has text boxes to allow you to enter a name, matric and course which are then applied to a new Student object when the Add Student button is clicked.
2. Add a third button called “Class List” and a ListBox control. Add an event handler to the “Class List” button which will access each student object and add the name of the student to the list box.
3. Add a button called “Clear” which will remove all students from the student list.
4. Draw a class diagram in Visual Studio to document your University system so far.
5. Lectures have the following attributes, title, tutor name and. Create a Lecture class within your system. There should be a 1:M relation between University and Lecture. Modify your form to contain Add Lecture and Show Lecture buttons and text boxes for the lecture attributes
6. Add code to your new buttons to allow lectures to be added, stored and displayed in a similar manner to students.
7. Update your class diagram to include the Lecture class.
8. **CHALLENGE:**  Can you create a 1:M relation between Lecture and Student that allows many students to be enrolled in a lecture? Can you build a GUI that allows students to be enrolled in a Lecture and a class list for a given lecture to be displayed? HINT – Work on your design using a class diagram first.

Show your attempt to your tutor for feedback.