LINQ TO XML

employee.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<Employees>
  <Employee>
    <EmpId>1</EmpId>
    <Name>Sam</Name>
    <Sex>Male</Sex>
    <Phone Type="Home">423-555-0124</Phone>
    <Phone Type="Work">424-555-0545</Phone>
    <Address>
      <Street>7A Cox Street</Street>
      <City>Acampo</City>
      <State>CA</State>
      <Zip>95220</Zip>
      <Country>USA</Country>
    </Address>
  </Employee>
  <Employee>
    <EmpId>2</EmpId>
    <Name>Lucy</Name>
    <Sex>Female</Sex>
    <Phone Type="Home">143-555-0763</Phone>
    <Phone Type="Work">434-555-0567</Phone>
    <Address>
      <Street>Jess Bay</Street>
      <City>Alta</City>
      <State>CA</State>
      <Zip>95701</Zip>
      <Country>USA</Country>
    </Address>
  </Employee>
  <Employee>
    <EmpId>3</EmpId>
    <Name>Kate</Name>
    <Sex>Female</Sex>
    <Phone Type="Home">166-555-0231</Phone>
    <Phone Type="Work">233-555-0442</Phone>
    <Address>
      <Street>23 Boxen Street</Street>
      <City>Milford</City>
      <State>CA</State>
      <Zip>96121</Zip>
      <Country>USA</Country>
    </Address>
  </Employee>
```

```
 <Employee>
  <EmpId>4</EmpId>
  <Name>Chris</Name>
  <Sex>Male</Sex>
  <Phone Type="Home">564-555-0122</Phone>
  <Phone Type="Work">442-555-0154</Phone>
  <Address>
   <Street>124 Kutbay</Street>
   <City>Montara</City>
   <State>CA</State>
   <Zip>94037</Zip>
   <Country>USA</Country>
  </Address>
 </Employee>
</Employees>
```

Code to process employee.xml

```
namespace linqtoxml
{
    class Program
    {
        static void Main(string[] args)
        {   //How Do I Read XML using LINQ to XML
            //using xelement

            XElement xelement = XElement.Load("employees.xml");
            IEnumerable<XElement> employees = xelement.Elements();
            // Read the entire XML

            foreach (var employee in employees)
            {
                Console.WriteLine(employee);
            }
            //Using xdocument

            XDocument xdocument = XDocument.Load("employees.xml");
            IEnumerable<XElement> employees1 = xdocument.Elements();
            foreach (var employee in employees1)
            {
                Console.WriteLine(employee);
            }
            //Accessing a Single Element using LINQ to XML

            Console.WriteLine("List of all Employee Names :");
            foreach (var employee in employees)
            {
                Console.WriteLine(employee.Element("Name").Value);
            }
            //Accessing Multiple Elements using LINQ to XML

            Console.WriteLine("List of all Employee Names along with their ID:");
            foreach (var employee in employees)
            {
```

```
                Console.WriteLine("{0} has Employee ID {1}",
                    employee.Element("Name").Value,
                    employee.Element("EmpId").Value);
            }
            //Accessing all Elements having a Specific Attribute using LINQ to XML

            var name = from nm in xelement.Elements("Employee")
                       where (string)nm.Element("Sex") == "Female"
                       select nm;
            Console.WriteLine("Details of Female Employees:");
            foreach (XElement xEle in name)
                Console.WriteLine(xEle);
            //Accessing Specific Element having a Specific Attribute using LINQ to XML

            var homePhone = from phoneno in xelement.Elements("Employee")
                            where (string)phoneno.Element("Phone").Attribute("Type") ==
"Home"
                            select phoneno;
            Console.WriteLine("List HomePhone Nos.");
            foreach (XElement xEle in homePhone)
            { Console.WriteLine(xEle.Element("Phone").Value);
            }
            //Finding an Element within another Element using LINQ to XML

            var addresses = from address in xelement.Elements("Employee")
                            where (string)address.Element("Address").Element("City") ==
"Alta"
                            select address;
            Console.WriteLine("Details of Employees living in Alta City");

            foreach (XElement xEle in addresses)
                Console.WriteLine(xEle);
            //Finding Nested Elements (using Descendants Axis) using LINQ to XML

            Console.WriteLine("List of all Zip Codes");
            foreach (XElement xEle in xelement.Descendants("Zip"))
            {  Console.WriteLine((string)xEle);
            }
            //Applying Sorting on Elements using LINQ to XML

            IEnumerable<string> codes = from code in xelement.Elements("Employee")
                                        let zip =
(string)code.Element("Address").Element("Zip")
                                        orderby zip
                                        select zip;
            Console.WriteLine("List and Sort all Zip Codes");
            foreach (string zp in codes)
                Console.WriteLine(zp);
            //Create an XML Document with Xml Declaration/Namespace/Comments using LINQ
to XML

            XNamespace empNM = "urn:lst-emp:emp";
            XDocument xDoc = new XDocument(
                        new XDeclaration("1.0", "UTF-16", null),
                        new XElement(empNM + "Employees",
                            new XElement("Employee",
                                new XComment("Only 3 elements for demo purposes"),
                                new XElement("EmpId", "5"),
```

```
                                new XElement("Name", "Kimmy"),
                                new XElement("Sex", "Female")
                                )));

            StringWriter sw = new StringWriter();
            xDoc.Save(sw);
            Console.WriteLine(sw);
            //Saving the XML Document to a XMLWriter or to the disk using LINQ to XML

            XmlWriter xWrite = XmlWriter.Create(sw);
            xDoc.Save(xWrite);
            xWrite.Close();

            // Save to Disk
            xDoc.Save("Something.xml");
            Console.WriteLine("Saved");
            //Loading an XML Document using XML Reader using LINQ to XML

            XmlReader xRead = XmlReader.Create(@"Employees.xml");
            XElement xEle1 = XElement.Load(xRead);
            Console.WriteLine(xEle1);
            xRead.Close();
            //Finding the Element Count based on a condition using LINQ to XML

            var stCnt = from address in xelement.Elements("Employee")
                        where (string)address.Element("Address").Element("State") == "CA"
                        select address;
            Console.WriteLine("No of Employees living in CA State are {0}",
stCnt.Count());
            //Adding a new Element at runtime using LINQ to XML

            XElement xEles = XElement.Load("Employees.xml");
            xEles.Add(new XElement("Employee",
                   new XElement("EmpId", 5),
                   new XElement("Name", "George")));
            Console.Write(xEles);
            //Adding an attribute to an Element using LINQ to XML

            xEles.Add(new XElement("Employee",
        new XElement("EmpId", 5),
        new XElement("Phone", "423-555-4224", new XAttribute("Type", "Home"))));
            Console.Write(xEles);
            //Replacing Contents of an Element/Elements using LINQ to XML

            var countries =
xEles.Elements("Employee").Elements("Address").Elements("Country").ToList();
            foreach (XElement cEle in countries)
                cEle.ReplaceNodes("United States Of America");
            Console.Write(xEles);
            //Deleting an Element based on a condition using LINQ to XML

            var addr = xEles.Elements("Employee").ToList();
            foreach (XElement addEle in addr)
                addEle.SetElementValue("Address", null);
            Console.Write(xEles);
            //Saving/Persisting Changes to the XML using LINQ to XML
            XElement xElems = XElement.Load("Employees.xml");
```

```
            xElems.Add(new XElement("Employee",
            new XElement("EmpId", 6),
            new XElement("Name", "George"),
            new XElement("Sex", "Male"),
            new XElement("Phone", "423-555-4224", new XAttribute("Type", "Home")),
            new XElement("Phone", "424-555-0545", new XAttribute("Type", "Work")),
            new XElement("Address",
                new XElement("Street", "Fred Park, East Bay"),
                new XElement("City", "Acampo"),
                new XElement("State", "CA"),
                new XElement("Zip", "95220"),
                new XElement("Country", "USA"))));

            xElems.Save("Employees.xml");

            Console.WriteLine(xElems);



            Console.ReadLine();



        }
    }
}
```