[illegible] LINQ

[illegible]

LINQ
Microsoft®
.NET
XML
</>

[illegible] XML [illegible]. [illegible]
XML [illegible]. [illegible] XML [illegible]. [illegible] XML [illegible].

[illegible] XML

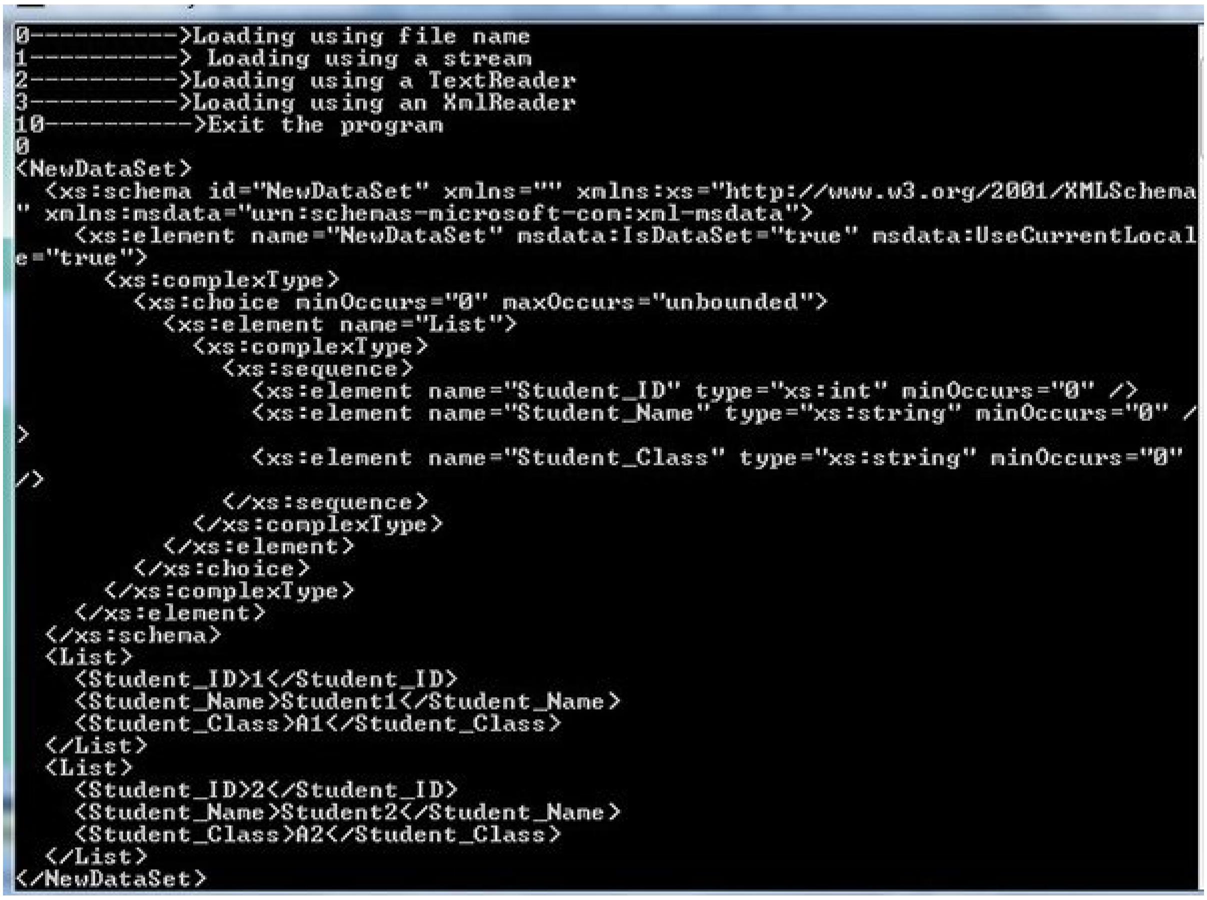
[illegible] XML [illegible].
System.Xml.Linq.XElement [illegible] XML [illegible]. [illegible]
XML [illegible]. XElement [illegible] XML [illegible]. [illegible] Load [illegible] XML [illegible]. XElement [illegible]. [illegible] XML [illegible] Load [illegible]. [illegible] Switch [illegible]. ([illegible] myList.xml [illegible].)

[illegible]:

```
using System;
using System.Xml;
using System.Xml.Linq;
using System.IO;
        static void Main(string[] args)
        {

            string filename = @”c:\myList.xml”;
            Byte m_case;
            XElement root;
            while (true)
            {
            Console.WriteLine(“0---------->Loading using file name”);
            Console.WriteLine(“1----------> Loading using a stream”);
            Console.WriteLine(“2---------->Loading using a TextReader”);
            Console.WriteLine(“3---------->Loading using an XmlReader”);
            Console.WriteLine(“10---------->Exit the program”);
            m_case= Convert.ToByte(Console.ReadLine());
            switch(m_case)
            {
                case 0:
            root = XElement.Load(filename);

            Console.WriteLine(root);
                    break;
                case 1:
            FileStream filestream = File.OpenRead(filename);
            root = XElement.Load(filestream);

            Console.WriteLine(root);
                    break;
                case 2:
                    TextReader reader = new StreamReader(filename);
                    root = XElement.Load(reader);
                    Console.WriteLine(root);
                    break;
                case 3:
                    XmlReader xmlreader = new XmlTextReader(new StreamReader(filename));
                    root = XElement.Load(xmlreader);
                    Console.WriteLine(root);
                    break;
                case 10:
                    return;
            }
        }
}
```

[illegible] myList.xml [illegible] C [illegible].
Switch [illegible].

[illegible] XML [illegible]
[illegible] 1 [illegible].



```
0---------->Loading using file name
1----------> Loading using a stream
2---------->Loading using a TextReader
3---------->Loading using an XmlReader
10---------->Exit the program
0
<NewDataSet>
  <xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema
" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
    <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:UseCurrentLocal
e="true">
      <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="List">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Student_ID" type="xs:int" minOccurs="0" />
                <xs:element name="Student_Name" type="xs:string" minOccurs="0" /
>
                <xs:element name="Student_Class" type="xs:string" minOccurs="0"
/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:schema>
  <List>
    <Student_ID>1</Student_ID>
    <Student_Name>Student1</Student_Name>
    <Student_Class>A1</Student_Class>
  </List>
  <List>
    <Student_ID>2</Student_ID>
    <Student_Name>Student2</Student_Name>
    <Student_Class>A2</Student_Class>
  </List>
</NewDataSet>
```

[illegible] -1 [illegible] XML [illegible].

[illegible] XML

[illegible] XML [illegible]. [illegible]
[illegible] XML [illegible].
[illegible] XML [illegible].

[illegible] where [illegible]. [illegible] select [illegible]. [illegible] orderby [illegible].

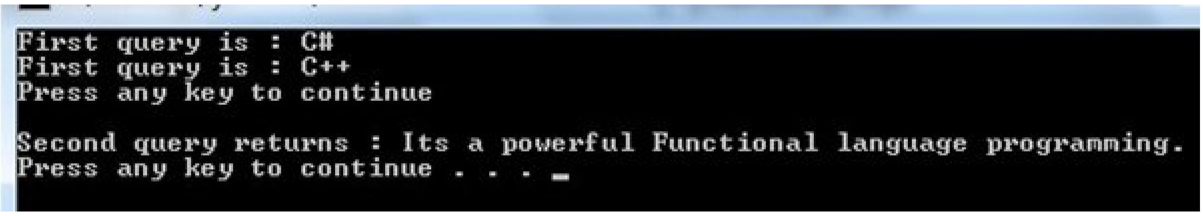
[illegible] XML [illegible] myxml.xml [illegible] Rate [illegible] 1 [illegible] LangaugeName [illegible] Rate [illegible] 2 [illegible] Describ [illegible] myxml.xml [illegible].

:[illegible]

<?xml version=”1.0” encoding=”utf-8”?>

```
<ComputersWorld>
  <Programming>
    <LanguageName>C#</LanguageName>
    <Describ>Its a powerful language programming of .Net’s family.</Describ>
    <Rate>1</Rate>
    <Rich>true</Rich>
  </Programming>
  <Programming>
    <LanguageName>C++</LanguageName>
    <Describ>Its a powerful language programming</Describ>
    <Rate>1</Rate>
    <Rich>true</Rich>
  </Programming>
  <Programming>
    <LanguageName>F#</LanguageName>
    <Describ>Its a powerful Functional language programming.</Describ>
    <Rate>2</Rate>
    <Rich>true</Rich>
  </Programming>
  <Database>
    <DatabaseName>SqlServer</DatabaseName>
    <Describ>Its a good and secure DBMS.</Describ>
    <Rate>1</Rate>
    <Rich>true</Rich>
  </Database>
  <Security>
    <OperatingSystem>Windowsr</OperatingSystem>
    <Describ>It has high security feature</Describ>
    <Rate>1</Rate>
    <Rich>true</Rich>
  </Security>
</ComputersWorld>
```

حال برای آنکه بتوانید دادههای درون فایل XML را مشاهده کنید دستوراتی که در فهرست سه مشاهده میکنید را در برنامه کاربردی خود وارد کنید . خروجی فهرست سه را در شکل 2 مشاهده میکنید .



```
First query is : C#
First query is : C++
Press any key to continue

Second query returns : Its a powerful Functional language programming.
Press any key to continue . . . _
```

شکل 2- جستوجوی مقادیر درون فایلهای *XML*.

فهرست سه:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Xml.Linq;
        static void Main(string[] args)
        {

            XElement rootElement = XElement.Load(@”c:\myxml.xml”);
                IEnumerable<string> query1 = from elem in rootElement.Elements()
where (elem.Name == “Programming” && ((string)elem.Element(“Rate”))== “1”)
                    select ((string)elem.Element(“LanguageName”));
                foreach (string str1 in query1)
                {
                    Console.WriteLine(“First query is : {0}”, str1);
                }

                Console.WriteLine(“Press any key to continue”);
                Console.ReadLine();

                IEnumerable<string> query2 = rootElement.Elements().Where(e
                => e.Name == “Programming”
                && (string)e.Element(“Rate”) == “2”).Select(
                e => (string)e.Element(“Describ”));
                foreach (string str1 in query2)
                {
        Console.WriteLine(“Second query returns : {0}”, str1);
                }
}
```

[illegible] XML [illegible]

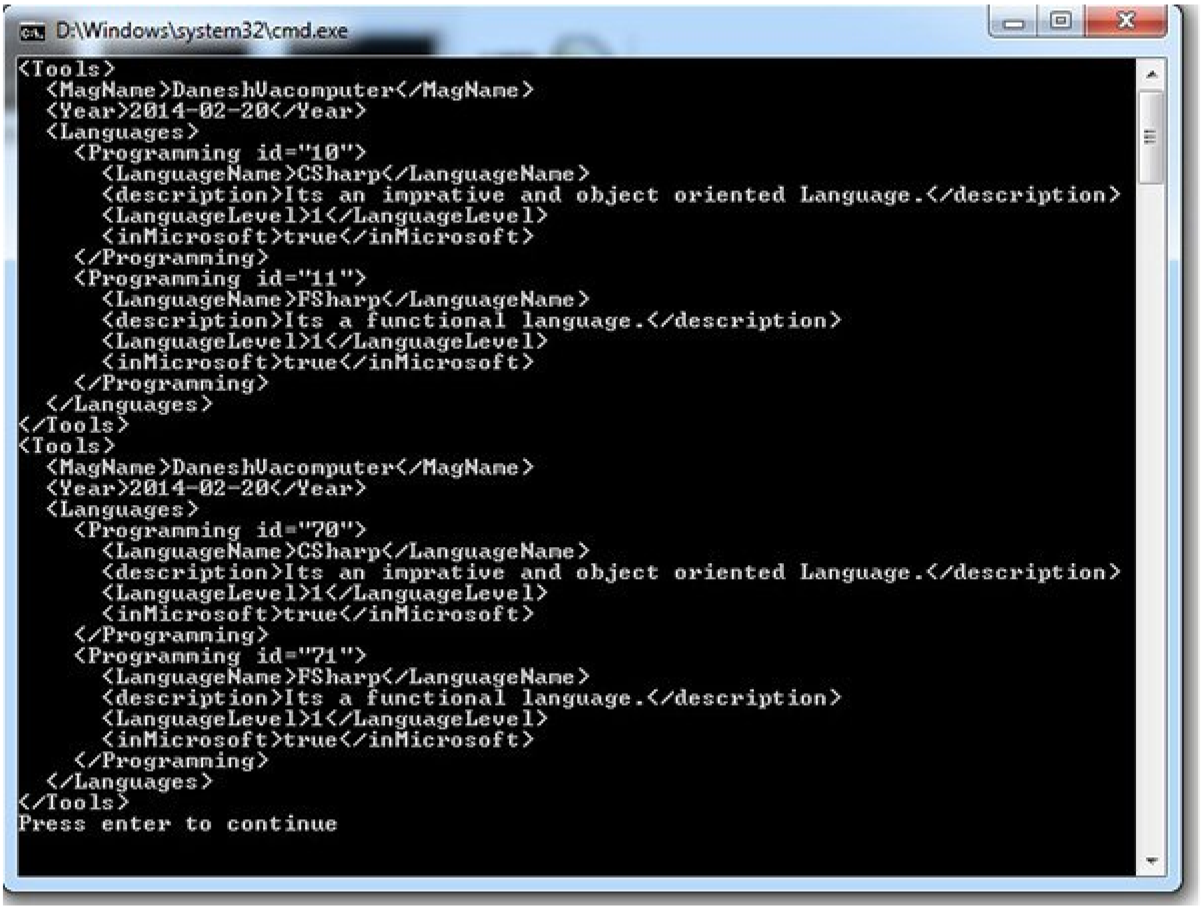
[illegible] XML [illegible]. [illegible] Add [illegible] Remove [illegible] XElement [illegible]. [illegible]. [illegible] Attribute [illegible] Element [illegible]. [illegible] ReplaceAttributes [illegible] ReplaceNodes [illegible]. [illegible] ReplaceWith [illegible]. [illegible] Remove [illegible] XElement [illegible]. [illegible] myxml.xml [illegible] C [illegible]. [illegible] xml [illegible].)

:[illegible]

```
<?xml version=”1.0” ?>
<!--An easy eaxmple of XML-->
<Tools>
<MagName>DaneshVacomputer</MagName>
<Year>2014-02-20</Year>
<Languages>
<Programming id=”10”>
<LanguageName>CSharp</LanguageName>
<description>Its an imprative and object oriented Language.</description>
<LanguageLevel>1</LanguageLevel>
<inMicrosoft>true</inMicrosoft>
</Programming>
<Programming id=”11”>
```

```
<LanguageName>FSharp</LanguageName>
<description>Its a functional language.</description>
<LanguageLevel>1</LanguageLevel>
<inMicrosoft>true</inMicrosoft>
</Programming>
</Languages>
</Tools>
```

[illegible] XML [illegible] 3 [illegible].



```
<Tools>
  <MagName>DaneshVacomputer</MagName>
  <Year>2014-02-20</Year>
  <Languages>
    <Programming id="10">
      <LanguageName>CSharp</LanguageName>
      <description>Its an imprative and object oriented Language.</description>
      <LanguageLevel>1</LanguageLevel>
      <inMicrosoft>true</inMicrosoft>
    </Programming>
    <Programming id="11">
      <LanguageName>FSharp</LanguageName>
      <description>Its a functional language.</description>
      <LanguageLevel>1</LanguageLevel>
      <inMicrosoft>true</inMicrosoft>
    </Programming>
  </Languages>
</Tools>
<Tools>
  <MagName>DaneshVacomputer</MagName>
  <Year>2014-02-20</Year>
  <Languages>
    <Programming id="70">
      <LanguageName>CSharp</LanguageName>
      <description>Its an imprative and object oriented Language.</description>
      <LanguageLevel>1</LanguageLevel>
      <inMicrosoft>true</inMicrosoft>
    </Programming>
    <Programming id="71">
      <LanguageName>FSharp</LanguageName>
      <description>Its a functional language.</description>
      <LanguageLevel>1</LanguageLevel>
      <inMicrosoft>true</inMicrosoft>
    </Programming>
  </Languages>
</Tools>
Press enter to continue
```

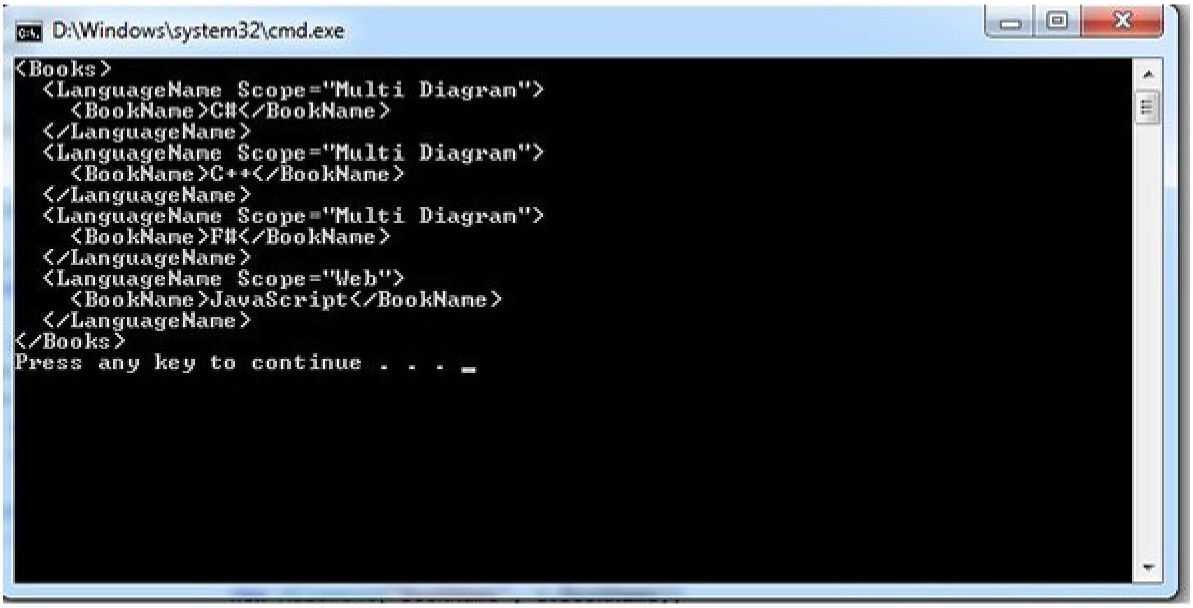
[illegible] 3- [illegible] XML [illegible].

[illegible]:

```
using System.Collections.Generic;
using System.Data.Linq.Mapping;
using System.Data.Linq;
using System.Linq;
using System;
using System.Xml;
using System.Xml.Linq;

namespace ConsoleApplication8
{
    class Program
    {
        static void Main(string[] args)
        {
            XElement root = XElement.Load(@”c:\myxml.xml”);
            Console.WriteLine(root);

            IEnumerable<XElement> Elements
            = from elem in root.Element(“Languages”).Elements()
              where (elem.Name == “Programming”)
              select elem;
            foreach (XElement elem in Elements)
            {
                int Value = Byte.Parse((string)elem.Attribute(“id”));
       elem.ReplaceAttributes(new XAttribute(“id”, Value + 60));
            }
            Console.WriteLine(root);
            Console.WriteLine(“Press enter to continue”);
            Console.ReadLine();
    IEnumerable<XElement> descrpElement = from elem in
                                                    root.Element(“Languages”).Elements()
                                                where (((string)elem.Element(“description”)).Contains(“Language”))
                                                select elem;
            foreach (XElement elem in descrpElement)
            {
                elem.Remove();
            }
            Console.WriteLine(root);
            Console.WriteLine(“Press enter to continue”);
            Console.ReadLine();

            XElement newElement = new XElement(“Programming”,
            new XAttribute(“id”, 12),
            new XElement(“LanguageName”, “C++ Language”),
            new XElement(“description”,
            “Ita an high level language”),
            new XElement(“inMicrosoft”, true));
            root.Element(“Languages”).Add(newElement);
            Console.WriteLine(root);
        }
    }
}
```

[illegible] XML [illegible]

[illegible] XML [illegible]
[illegible] XML [illegible]
[illegible] ADO.Net [illegible]
[illegible]
[illegible] LINQ to XML [illegible]
[illegible]
[illegible] XElement [illegible] XML [illegible]
[illegible] 4 [illegible]



```
<Books>
  <LanguageName Scope="Multi Diagram">
    <BookName>C#</BookName>
  </LanguageName>
  <LanguageName Scope="Multi Diagram">
    <BookName>C++</BookName>
  </LanguageName>
  <LanguageName Scope="Multi Diagram">
    <BookName>F#</BookName>
  </LanguageName>
  <LanguageName Scope="Web">
    <BookName>JavaScript</BookName>
  </LanguageName>
</Books>
Press any key to continue . . .
```

[illegible] 4- [illegible] XML [illegible].

[illegible]:

```
using System.Data.Linq.Mapping;
using System.Data.Linq;
using System.Linq;
using System;
using System.Xml;
using System.Xml.Linq;
      static void Main(string[] args)
      {

var Languages = new[] {
new { BookName = “C#”, Scope = “Multi Diagram”},
new { BookName = “C++”,Scope = “Multi Diagram” },
new { BookName  = “F#”, Scope = “Multi Diagram” },
new { BookName  = “JavaScript”,Scope = “Web”}
};
XElement newelement =
new XElement(“Books”, from c in Languages
select new XElement(“LanguageName”, new XAttribute(“Scope”, c.Scope),
new XElement(“BookName”, c.BookName))
);
Console.WriteLine(newelement);
}
```

[illegible] Linq to XML [illegible].

[illegible]:

[illegible]

[illegible]:

[illegible]

:□□□□□□□ □□□□□

10:35 - 12/02/1396

:□□□□□□

□□□□□□□□□□□□□ □□□□□□ - □□□□□□□□□□□□ - □□□□□□□ □□□□□□□□ - LINQ

□□□□□□

https://www.shabakeh-mag.com/workshop/7707/%D8%AA%DA%A9%D9%86%DB%8C%DA%A9%E2%80%8C%D9%87%D8%A7%DB%8C-%DA%A9%D8%A7%D8%B1%D8%A8%D8%B1%D8%AF%DB%8C-%D8%A7%D8%B3%D8%AA%D8%AE%D8%B1%D8%A7%D8%AC-%D8%AF%D8%A7%D8%AF%D9%87%E2%80%8C%D9%87%D8%A7-%D8%A8%D8%A7-%D9%84%DB%8C%D9%86%DA%A9-%D8%A8%D8%AE%D8%B4-%DA%86%D9%87%D8%A7%D8%B1%D9%85:□□□□