

1. ถ้ามีเอกสาร XML ชื่อ student.xml ดังต่อไปนี้

ให้สร้างเอกสาร student.xslt โดยการใช้คำสั่ง <xsl:apply-templates> โดยการเติม code ในส่วนที่ขาดหายไปของเอกสาร student.xslt ซึ่งจะมีการเรียงลำดับรหัสนักศึกษาจากน้อยไปมาก และสำหรับนักศึกษาแต่ละคนให้เรียงลำดับวิชา (course) ที่นศ.ได้ลงด้วยจากรหัสวิชามากไปน้อยเพื่อให้ได้ผลลัพธ์ดังแสดงในภาพที่ 1

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="student.xslt" ?>
<Students>
  <Student sid="S00103">
    <name>
      <first>Somsak</first>
      <last>Sandee</last>
    </name>
    <courses>
      <course cid="CS001" credit="3" />
      <course cid="CS004" credit="3" />
    </courses>
    <telephone>2323232</telephone>
  </Student>
  <Student sid="S00102">
    <name>
      <first>Wirai</first>
      <last>Mesuk</last>
    </name>
    <courses>
      <course cid="CS005" credit="3" />
      <course cid="CS007" credit="3" />
    </courses>
    <telephone>8662551</telephone>
  </Student>
</Students>
```

student.xml

Student Information for student ID: S00102

FirstName: Wirai **LastName:** Mesuk
Telephone: 8662551

Couse ID	Credit
CS007	3
CS005	3

Student Information for student ID: S00103

FirstName: Somsak **LastName:** Sandee
Telephone: 2323232

Couse ID	Credit
CS004	3
CS001	3

ภาพที่ 1 แสดงผลลัพธ์ที่เกิดจากการประมวลผลเอกสาร student.xml และ student.xslt

```

<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xsl:output method="html" version="1.0" encoding="UTF-8" indent="yes"/>
<xsl:template match="/">
  <html>
    <body>
      <xsl:apply-templates select=".....">
        .....
      </xsl:apply-templates>
    </body>
  </html>
</xsl:template>

<xsl:template match=".....">
  <h3>Student Information for student ID: .....</h3>
  .....
  .....
  <table border="1">
    <th>Course ID</th> <th>Credit</th>
    <xsl:apply-templates select="....." >
      .....
    </xsl:apply-templates>
  </table>
</xsl:template>

<xsl:template match=".....">
  <tr>
    .....
    .....
  </tr>
</xsl:template>

</xsl:stylesheet>

```

student.xslt

3. จากเอกสาร books.xml ไฟล์ต่อไปนี้

```

<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="books.xsl" ?>
<books>
  <book id="_23234322">
    <name>XML Technology</name>
    <authors>
      <author>
        <firstName>Smith</firstName>
        <lastName>Jason</lastName>
      </author>
      <author>
        <firstName>Robert</firstName>
        <lastName>Lee</lastName>
      </author>
    </authors>
    <publisher>Thomsons</publisher>
    <price>30$</price>
  </book>
</books>

```

3.1 จงแสดงคำสั่งของ DOM ในการเข้าถึงข้อมูล lastName ที่อยู่ภายใต้ element <author> โดย

3.1.1 ใช้ property ของ document object และ node object ในการเข้าถึงข้อมูล “Lee” (7 คะแนน)

3.1.2 ใช้ method ของ document object ร่วมกับ property ของ node object ในการเข้าถึงข้อมูล “Jacson” (3 คะแนน)

(ให้เพิ่มเติมเฉพาะ code ในส่วนที่ยังขาดอยู่ ไม่ต้องลอก code ทั้งหมดในตัวอย่างมาแสดง)

```
<HTML>
  <SCRIPT LANGUAGE="JavaScript" for="window" event="onload">
    let text= "<books>" +
      "<book id='_232343222'>" +
        "<name>XML Technology</name>" +
        "<authors>" +
          "<author>" +
            "<firstName>Smith</firstName>" +
            "<lastName>Jacson</lastName>" +
          "</author>" +
          "<author>" +
            "<firstName>Robert</firstName>" +
            "<lastName>Lee</lastName>" +
          "</author>" +
        "</authors>" +
        "<publisher>Thomsons</publisher>" +
        "<price>30$</price>" +
      "</book>" +
    "</books>";

    let parser = new DOMParser()
    let xmlDoc = parser.parseFromString(text, "text/xml")
    alert(xmlDoc.documentElement.firstChild.childNodes[1].childNodes[1].childNodes[1].firstChild.nodeValue);
    alert(xmlDoc.getElementsByTagName("lastName")[0].firstChild.nodeValue);
  </SCRIPT>
</HTML>
```

3.1.1

3.1.2

3.2 ให้เขียนส่วนของคำสั่งเพื่อ

3.2.1 ทำการเพิ่ม element <midName> ต่อจาก element <firstName> เข้าไปในเรคคอร์ดของ <author> แต่ละ record พร้อมทั้งใส่ข้อมูลสำหรับ element <midName> แต่ละ element นี้ด้วย (10 คะแนน)

Note: ข้อนี้ณ.จะต้องเลือกใช้ method ของ node object คือ *parent.insertBefore(new, old);* เพื่อทำการแทรก (insert) element <midName> ซึ่งเป็น node ที่ถูกสร้างขึ้นใหม่ (new node) ให้ไปอยู่ก่อน element <lastName> ซึ่งถือว่าเป็น old node ที่เราต้องการให้แทรก ซึ่งทุก elements นี้อยู่ภายใต้ parent เดียวกันคือ <author>

```

<HTML>
  <SCRIPT LANGUAGE="JavaScript" for="window" event="onload">
    let text= "<books>" +
      "<book id='_232343222'>" +
        "<name>XML Technology</name>" +
        "<authors>" +
          "<author>" +
            "<firstName>Smith</firstName>" +
            "<lastName>Jason</lastName>" +
          "</author>" +
          "<author>" +
            "<firstName>Robert</firstName>" +
            "<lastName>Lee</lastName>" +
          "</author>" +
        "</authors>" +
        "<publisher>Thomsons</publisher>" +
        "<price>30$</price>" +
      "</book>" +
    "</books>";

    let parser = new DOMParser()
    let xmldoc = parser.parseFromString(text, "text/xml")

    var auth1 = xmldoc.getElementsByTagName("author")[0];
    var auth2 = xmldoc.getElementsByTagName("author")[1];
    var last1 = xmldoc.getElementsByTagName("lastName")[0];
    var last2 = xmldoc.getElementsByTagName("lastName")[1];

    var midName1 = xmldoc.createElement("midName");
    var midNametxt1 = xmldoc.createTextNode(" Syl ");
    midName1.appendChild(midNametxt1);
    var midName2 = xmldoc.createElement("midName");
    var midNametxt2 = xmldoc.createTextNode(" Jam ");
    midName2.appendChild(midNametxt2);

    auth1.insertBefore(midName1,last1);
    auth2.insertBefore(midName2,last2);

    let Xstr = new XMLSerializer().serializeToString(xmldoc);
    alert(Xstr);

  </SCRIPT>

```

localhost บอกรหัส

```

<books> <book id="_232343222"> <name>XML Technology</
name> <authors> <author> <firstName>Smith</
firstName> <midName> Syl </midName> <lastName>Jason</
lastName> </author> <author> <firstName>Robert</
firstName> <midName> Jam </midName> <lastName>Lee</
lastName> </author> </authors> <publisher>Thomsons</
publisher> <price>30$</price> </book> </books>

```

ปิดกล่อง

3.2.2 เพิ่ม record author อีก 1 record ที่มีโครงสร้างของ element ต่างๆเหมือนกับ 2 records แรกนี้โดยใช้วิธีการ clone node และให้เปลี่ยนแปลงแก้ไขข้อมูลภายในแต่ละ element สุดท้ายให้นำ recode ใหม่นี้ไปต่อเข้ากับ element <authors> ด้วย (8 คะแนน)

```
<HTML>
  <SCRIPT LANGUAGE="JavaScript" for="window" event="onload">
    let text= "<books>"+
      "<book id=' 232343222'>"+
        "<name>XML Technology</name>"+
        "<authors>"+
          "<author>"+
            "<firstName>Smith</firstName>"+
            "<lastName>Jason</lastName>"+
          "</author>"+
          "<author>"+
            "<firstName>Robert</firstName>"+
            "<lastName>Lee</lastName>"+
          "</author>"+
        "</authors>"+
        "<publisher>Thomsons</publisher>"+
        "<price>30$</price>"+
      "</book>"+
    "</books>";
    let parser = new DOMParser()
    let xmldoc = parser.parseFromString(text, "text/xml")

    var auths = xmldoc.documentElement.firstChild.childNodes[1];
    var firstAuth = xmldoc.getElementsByTagName("author")[0];

    var cloneAuth = firstAuth.cloneNode(true);
    cloneAuth.firstChild.firstChild.nodeValue = "Ngamnij";
    cloneAuth.lastChild.firstChild.nodeValue="Arch-int";
    auths.appendChild(cloneAuth);

    let Xstr = new XMLSerializer().serializeToString(xmldoc);
    alert(Xstr);

  </SCRIPT>
</HTML>
```

localhost บอกรว่า

```
<books> <book id=" 232343222"> <name>XML Technology</
name> <authors> <author> <firstName> Smith</
firstName> <lastName> Jason</lastName> </
author> <author> <firstName> Robert</firstName> <lastName> Lee</
lastName> </author> <author> <firstName> Ngamnij</
firstName> <lastName> Arch-int</lastName> </author> </
authors> <publisher> Thomsons</publisher> <price> 30$</price> </
book> </books>
```

ตกลง