REG NO. : 412422205009

NAME : AKASH RAJ L

SOURCE CODE :

akash.xml

```
<employees>
   <employee>
      <id>101</id>
      <name>John Doe</name>
      <role>Developer</role>
      <department>Engineering</department>
      <salary>60000</salary>
   </employee>
   <employee>
      <id>102</id>
      <name>Jane Smith</name>
      <role>Manager</role>
      <department>Marketing</department>
      <salary>75000</salary>
   </employee>
   <employee>
      <id>103</id>
      <name>Robert Brown</name>
      <role>Analyst</role>
      <department>Finance</department>
      <salary>55000</salary>
   </employee>
   <employee>
      <id>104</id>
      <name>Emily White</name>
      <role>Designer</role>
      <department>Creative</department>
```

```
        <salary>50000</salary>
    </employee>
    <employee>
        <id>105</id>
        <name>Michael Johnson</name>
        <role>Sales Executive</role>
        <department>Sales</department>
        <salary>45000</salary>
    </employee>
</employees>
```

DOMParser1.java

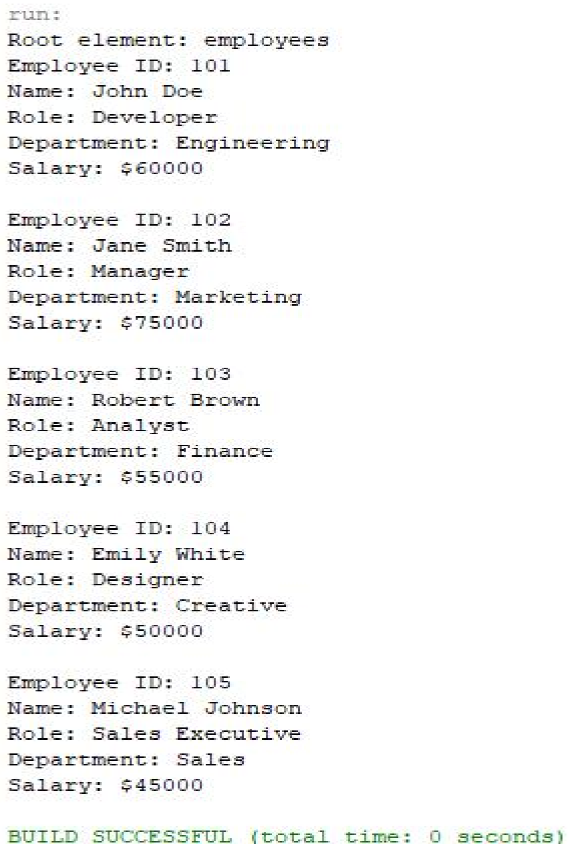
```
import javax.xml.parsers.DocumentBuilder;
import javax.xml.parsers.DocumentBuilderFactory;
import org.w3c.dom.*;

public class DOMParser1 {
    public static void main(String[] args) {
        try {
            // Initialize DocumentBuilder
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();

            // Load XML file using ClassLoader
            ClassLoader classLoader = DOMParser1.class.getClassLoader();
            Document document = builder.parse(classLoader.getResourceAsStream("akash.xml"));

            // Normalize XML structure
            document.getDocumentElement().normalize();

            // Root element
            System.out.println("Root element: " + document.getDocumentElement().getNodeName());
            // Parse employee nodes
            NodeList nodeList = document.getElementsByTagName("employee");
            for (int i = 0; i < nodeList.getLength(); i++) {
                Node node = nodeList.item(i);
                if (node.getNodeType() == Node.ELEMENT_NODE) {
                    Element employee = (Element) node;
                    System.out.println("Employee ID: " +
            employee.getElementsByTagName("id").item(0).getTextContent());
                    System.out.println("Name: " +
            employee.getElementsByTagName("name").item(0).getTextContent());
                    System.out.println("Role: " +
            employee.getElementsByTagName("role").item(0).getTextContent());
                    System.out.println("Department: " +
            employee.getElementsByTagName("department").item(0).getTextContent());
                    System.out.println("Salary: $" +
            employee.getElementsByTagName("salary").item(0).getTextContent());
                    System.out.println();
                }
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

SAXParser1.java

```
import org.xml.sax.Attributes;
import org.xml.sax.SAXException;
import org.xml.sax.helpers.DefaultHandler;
import javax.xml.parsers.SAXParser;
import javax.xml.parsers.SAXParserFactory;
import java.io.InputStream;

public class SAXParser1 {
    public static void main(String[] args) {
        try {
            SAXParserFactory factory = SAXParserFactory.newInstance();
            SAXParser saxParser = factory.newSAXParser();

            DefaultHandler handler = new DefaultHandler() {
                boolean bId = false;
                boolean bName = false;
                boolean bRole = false;
                boolean bDepartment = false;
                boolean bSalary = false;

                // Start element event
                public void startElement(String uri, String localName, String qName, Attributes attributes)
throws SAXException {
                    switch (qName) {
                        case "id" -> bId = true;
                        case "name" -> bName = true;
                        case "role" -> bRole = true;
                        case "department" -> bDepartment = true;
                        case "salary" -> bSalary = true;
                    }
                }

                // Characters event
                public void characters(char[] ch, int start, int length) throws SAXException {
                    if (bId) {
                        System.out.println("Employee ID: " + new String(ch, start, length));
                        bId = false;
                    } else if (bName) {
                        System.out.println("Name: " + new String(ch, start, length));
                        bName = false;
                    } else if (bRole) {
                        System.out.println("Role: " + new String(ch, start, length));
                        bRole = false;
                    } else if (bDepartment) {
                        System.out.println("Department: " + new String(ch, start, length));
                        bDepartment = false;
                    } else if (bSalary) {
                        System.out.println("Salary: $" + new String(ch, start, length));
                        bSalary = false;
                    }
                }
                // End element event
                public void endElement(String uri, String localName, String qName) throws
SAXException {
                    if (qName.equalsIgnoreCase("employee")) {
                        System.out.println();
                    }
                }
            };

            // Load XML file using ClassLoader as InputStream
            InputStream xmlInput =
SAXParser1.class.getClassLoader().getResourceAsStream("akash.xml");
            if (xmlInput == null) {
                throw new IllegalArgumentException("File not found!");
            }
            saxParser.parse(xmlInput, handler);

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

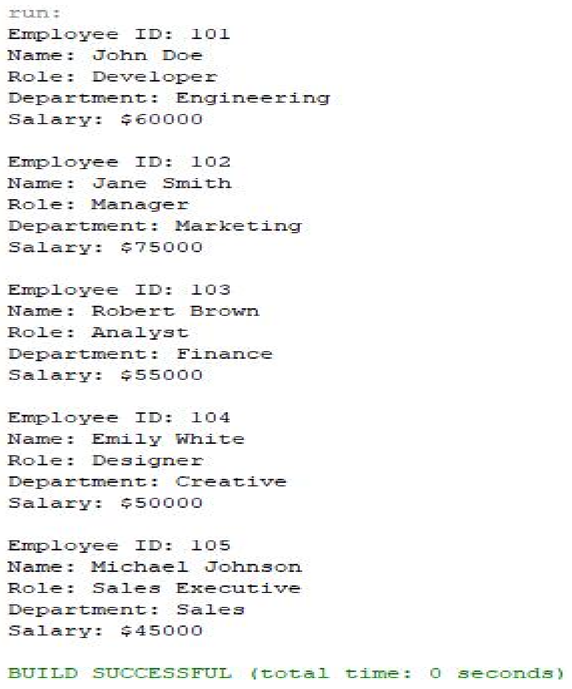
OUTPUT :

DOMParser



```
run:
Root element: employees
Employee ID: 101
Name: John Doe
Role: Developer
Department: Engineering
Salary: $60000

Employee ID: 102
Name: Jane Smith
Role: Manager
Department: Marketing
Salary: $75000

Employee ID: 103
Name: Robert Brown
Role: Analyst
Department: Finance
Salary: $55000

Employee ID: 104
Name: Emily White
Role: Designer
Department: Creative
Salary: $50000

Employee ID: 105
Name: Michael Johnson
Role: Sales Executive
Department: Sales
Salary: $45000

BUILD SUCCESSFUL (total time: 0 seconds)
```

SAXParser



```
run:
Employee ID: 101
Name: John Doe
Role: Developer
Department: Engineering
Salary: $60000

Employee ID: 102
Name: Jane Smith
Role: Manager
Department: Marketing
Salary: $75000

Employee ID: 103
Name: Robert Brown
Role: Analyst
Department: Finance
Salary: $55000

Employee ID: 104
Name: Emily White
Role: Designer
Department: Creative
Salary: $50000

Employee ID: 105
Name: Michael Johnson
Role: Sales Executive
Department: Sales
Salary: $45000

BUILD SUCCESSFUL (total time: 0 seconds)
```