# Parser Assignment

DOM Parser Case Study – Student Report Up

### **Scenario:**

You are given a file named Students.xml containing a list of student records. Each <student> element includes details such as name, age, department, marks, and grade.

Your task is to:

1. Parse the XML using Java DOM parser.
2. Display all students from the "Computer" department.
3. Identify students who scored below 40 marks and update their <grade> to "Fail".
4. Save the updated content into a new XML file called Students\_Updated.xml, ensuring proper formatting and indentation.

**Student.XML**

<students>

<student>

<name>Aryan</name>

<age>20</age>

<department>Computer</department>

<marks>78</marks>

<grade>A</grade>

</student>

<student>

<name>Neha</name>

<age>21</age>

<department>Electrical</department>

<marks>34</marks>

<grade>C</grade>

</student>

<student>

<name>Mehul</name>

<age>22</age>

<department>Computer</department>

<marks>89</marks>

<grade>A</grade>

</student>

</students>

**Objectives:**

 Load and parse Students.xml.

 Traverse each <student> node.

 Print names of students in "Computer" department.

 Update <grade> to "Fail" where <marks> is less than 40.

 Write modified data to Students\_Updated.xml.

### **Assignment Tasks:**

#### Task 1: Setup

* Import required Java XML packages.
* Load the XML file using DocumentBuilderFactory.

#### Task 2: Traverse and Display

* Loop through each <student> node.
* Print name and department of students in Computer department.

#### Task 3: Modify XML

* For students with <marks> less than 40, change <grade> to "Fail" using setTextContent().

#### Task 4: Save Output

* Write the modified DOM back to Students\_Updated.xml using Transformer.
* Set output formatting with indentation (OutputKeys.INDENT = "yes").

**Expected Result :**

Students in Computer Department:

- Aryan

- Mehul

Grade updated for students with marks < 40.

Updated file saved as Students\_Updated.xml

## **Assignment: Parse XML using SAX Parser – Case Study: Employee Directory**

### \*\*Objective:

Create a SAX-based Java application to parse an XML file that contains details of employees in a company. The application should display each employee’s information in a readable format.

Employee.xml

<employees>

<employee>

<id>101</id>

<name>Amit Sharma</name>

<department>Engineering</department>

<salary>75000</salary>

</employee>

<employee>

<id>102</id>

<name>Priya Mehra</name>

<department>HR</department>

<salary>55000</salary>

</employee>

<employee>

<id>103</id>

<name>Ravi Patel</name>

<department>Marketing</department>

<salary>60000</salary>

</employee>

</employees>

### **Task Requirements:**

1. Create a Java class EmployeeHandler by extending DefaultHandler.
2. Override the startElement, characters, and endElement methods.
3. In the characters() method, extract and display employee data.
4. In the main() method, configure the SAX parser and parse the file.
5. Format output like:

Employee ID: 101

Name : Amit Sharma

Department : Engineering

Salary : 75000

-----------------------------