**Lab Assignment 1- Working with Drools LHS (Left-Hand Side) Conditions**

**Objective:**

To develop and apply Drools rules using the Left-Hand Side (LHS) conditions effectively. You will create rules to manage a basic inventory and order processing system for an e-commerce platform, where discounts and special processing are applied based on customer type, order value, and product category.

**Problem Statement:**

You are tasked with creating a rule-based system for an e-commerce platform. The system should be capable of processing orders and applying specific business rules based on customer types, order amounts, and product categories. The rules will include conditions to determine discounts for different customer types, high-value orders, and special processing for specific product categories. The focus is on defining these conditions correctly on the Left-Hand Side (LHS) of the rules.

**Scenario:**

**Customer Types:**

* VIP: Eligible for a 20% discount if the order amount exceeds $500.
* Regular: Eligible for a 10% discount if the order amount exceeds $200.
* New: No discount is provided, but the order should still be processed.

**Product Categories:**

* Electronics: Requires special handling if the order amount exceeds $1000.
* Clothing: No special handling, but a flat 5% discount is applied if the order amount exceeds $150.

**Order Processing:**

* Every order must be marked as processed after applying the relevant rules.

**Assignment Tasks:**

**1. Setup the Drools Project**

* Create a new Drools project.
* Configure the project with necessary dependencies, ensuring Drools is integrated with your IDE (e.g., Eclipse, IntelliJ IDEA).

**2. Create the Fact Classes**

**Order.java:**

public class Order {

private String customerType;

private double orderAmount;

private String productCategory;

private boolean isProcessed;

public Order(String customerType, double orderAmount, String productCategory) {

this.customerType = customerType;

this.orderAmount = orderAmount;

this.productCategory = productCategory;

this.isProcessed = false;

}

public String getCustomerType() {

return customerType;

}

public void setCustomerType(String customerType) {

this.customerType = customerType;

}

public double getOrderAmount() {

return orderAmount;

}

public void setOrderAmount(double orderAmount) {

this.orderAmount = orderAmount;

}

public String getProductCategory() {

return productCategory;

}

public void setProductCategory(String productCategory) {

this.productCategory = productCategory;

}

public boolean isProcessed() {

return isProcessed;

}

public void setProcessed(boolean isProcessed) {

this.isProcessed = isProcessed;

}

@Override

public String toString() {

return "Order [customerType=" + customerType + ", orderAmount=" + orderAmount +

", productCategory=" + productCategory + ", isProcessed=" + isProcessed + "]";

}

}

**3. Write the DRL (Drools Rule Language) File**

**orderRules.drl:**

package com.example.rules;

import com.example.Order;

// Rule 1: Apply 20% discount for VIP customers on orders over $500

rule "VIP Customer Discount"

when

$order : Order(customerType == "VIP", orderAmount > 500)

then

System.out.println("Applying 20% discount for VIP customer");

$order.setOrderAmount($order.getOrderAmount() \* 0.8);

$order.setProcessed(true);

end

**4. Create the Main Class**

DroolsTest.java:

package com.example;

import org.kie.api.runtime.KieContainer;

import org.kie.api.runtime.KieSession;

import org.kie.api.KieServices;

public class DroolsTest {

public static void main(String[] args) {

try {

KieServices ks = KieServices.Factory.get();

KieContainer kContainer = ks.getKieClasspathContainer();

KieSession kSession = kContainer.newKieSession("ksession-rules");

// Create some orders

Order order1 = new Order("VIP", 600, "Electronics");

Order order2 = new Order("Regular", 250, "Clothing");

Order order3 = new Order("New", 100, "Books");

// Insert orders into the session

kSession.insert(order1);

kSession.insert(order2);

kSession.insert(order3);

// Fire all rules

kSession.fireAllRules();

// Print the final state of the orders

System.out.println(order1);

System.out.println(order2);

System.out.println(order3);

kSession.dispose();

} catch (Exception e) {

e.printStackTrace();

}

}

}

**5. Run the Application**

* Run the DroolsTest class.
* Observe the output in the console. Each order should be processed according to the rules defined in the .drl file.

**Expected Output:**

Applying 20% discount for VIP customer

Special handling for high-value Electronics order

Applying 10% discount for Regular customer

Applying 5% discount for Clothing

No discount applied for New customer

Processing order: Order [customerType=VIP, orderAmount=480.0, productCategory=Electronics, isProcessed=true]

Processing order: Order [customerType=Regular, orderAmount=213.75, productCategory=Clothing, isProcessed=true]

Processing order: Order [customerType=New, orderAmount=100.0, productCategory=Books, isProcessed=true]