Assignment -1

Student Name/ID Number:	Chathushi Jayarathna
Academic Year:	2022/23
Unit Assessor:	MS. Aravinder Kaur
Project Title:	Assignment1– **Inversion of Control**
Issue Date:	23/11/2022
Submission Date:	02/01/2023
Internal Verifier Name:	
Date:	02/01/2023

Learner declaration
I certify that the work submitted for this assignment is my own and research sources are fully acknowledged. Student signature: **Chathushi** Date: 02/01/2023

Module No:	1	**IU No:**	1	**Exercise No.**	1

Lab Assessment Statement

Assignment 1 - Inversion of Control

You are handling the Billing module of an e-commerce application. You have developed below class.

```
package lithan.training.javawebapp

public class BillingService {
  private PaymentProcessor paymentProcessor;

  public bool completePayment() {
    return paymentProcessor.process();
  }
}
```

Note:

- **You are using a 3rd party library (external library) to process the payments**
- **This external library provides an interface PaymentProcessor for clients to process the payments**
- **The external library supports 2 modes of payment. "gpay" and "credit_card".**
- **It also has a Factory class which will instantiate appropriate processor based on mode of payment.**
- **Below is the brief outline of the classes provided by external library**

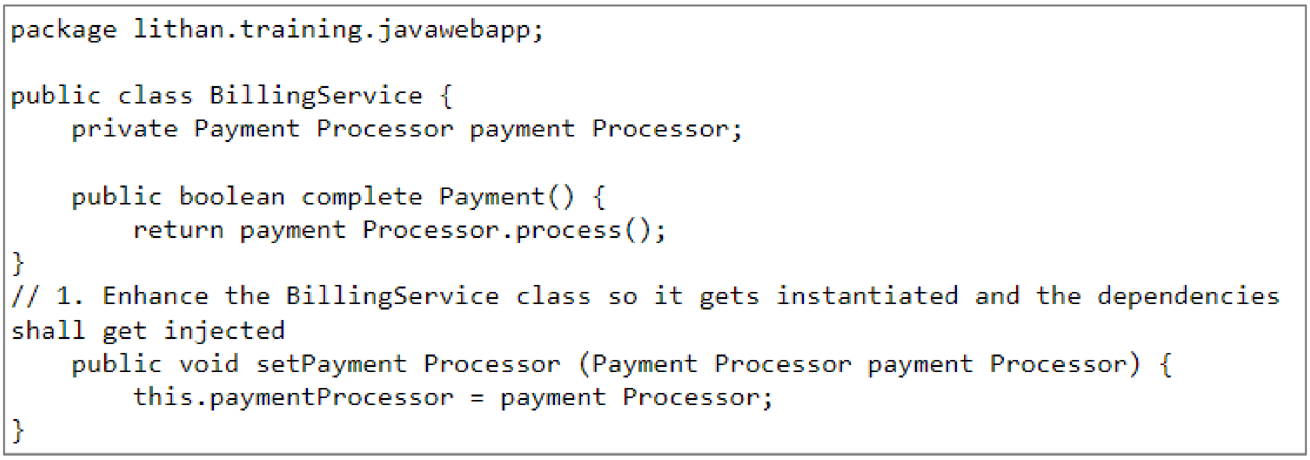
```
public interface PaymentProcessor {
  public bool process();
}

public class GooglePayProcessor implements PaymentProcessor {
  public bool process() {
    // process payment
    return true;
  }
}
```

```
public class CreditCardProcessor implements PaymentProcessor {
  public bool process() {
    // process payment
    return true;
  }
}

public PaymentProcessorFactory {
  private final GooglePayProcessor gpayProcessor = new
GooglePayProcessor();
  private final CreditCardProcessor cardProcessor = new CreditCardProcessor
();

  public PaymentProcessor getPaymentProcessor(String mode) {
    if ("gpay".equalsIgnoreCase(mode)) {
      return gpayProcessor;
    } else if ("credit_card".equalsIgnoreCase(mode)) {
      return cardProcessor ;
    }
    return null;
  }
}
```

1. **Enhance the BillingService class so it gets instantiated and the dependencies shall get injected.**
    - Hint: What method would you use to instantiate BillingService class? Remember, this is the bean developed by you.
2. **Write Java Configuration class to instantiate required classes from external library.**

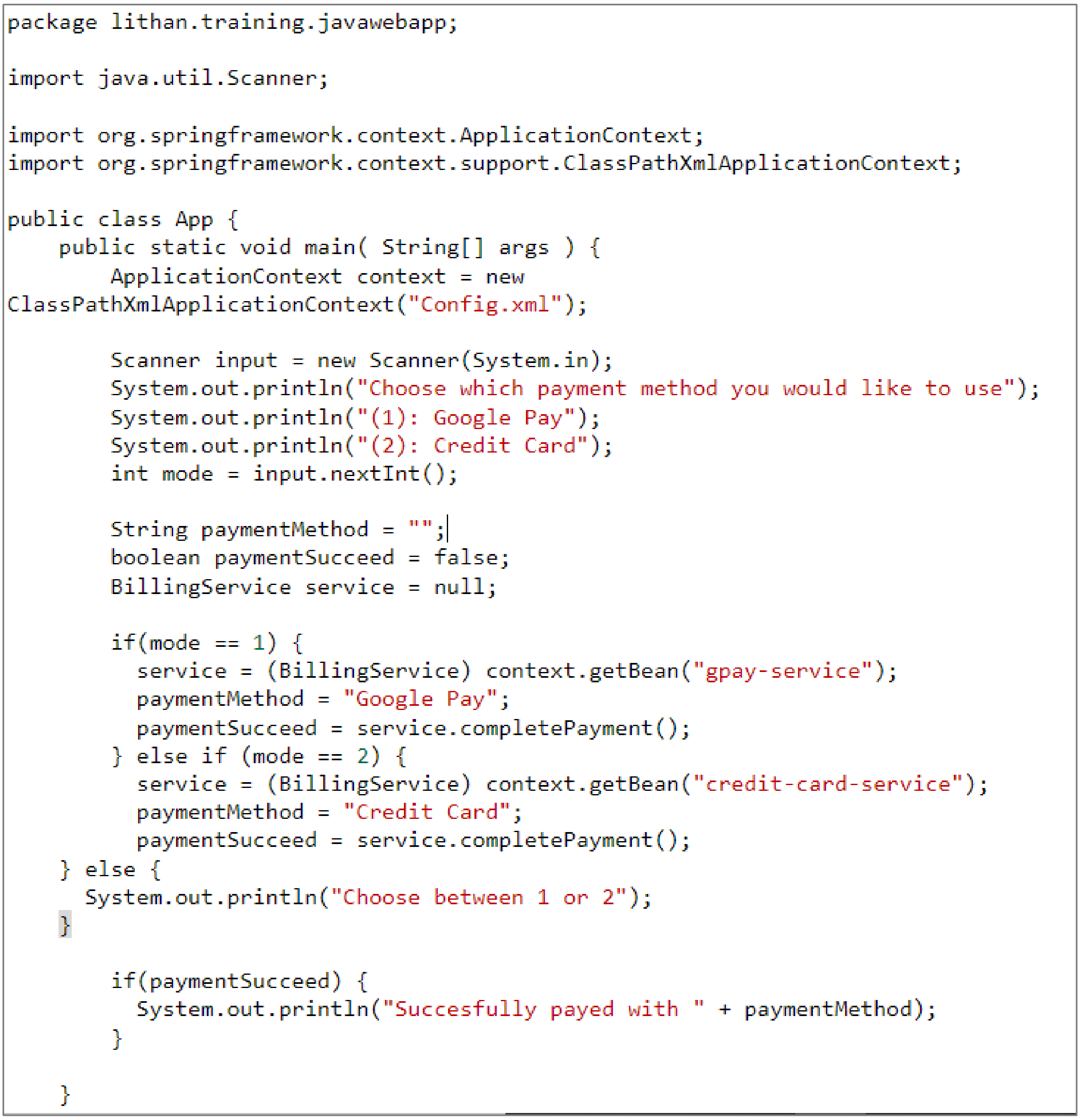
Technical Environment	-
Guidelines	-
Duration	120 ns

- ✓ **BillingService Class**



```
package lithan.training.javawebapp;

public class BillingService {
    private Payment Processor payment Processor;

    public boolean complete Payment() {
        return payment Processor.process();
}
// 1. Enhance the BillingService class so it gets instantiated and the dependencies
shall get injected
    public void setPayment Processor (Payment Processor payment Processor) {
        this.paymentProcessor = payment Processor;
}
```

- ✓ **APP.java**

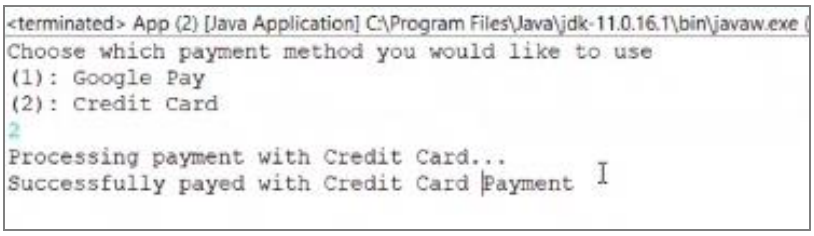


```
package lithan.training.javawebapp;

import java.util.Scanner;

import org.springframework.context.ApplicationContext;
import org.springframework.context.support.ClassPathXmlApplicationContext;

public class App {
    public static void main( String[] args ) {
        ApplicationContext context = new
ClassPathXmlApplicationContext("Config.xml");

        Scanner input = new Scanner(System.in);
        System.out.println("Choose which payment method you would like to use");
        System.out.println("(1): Google Pay");
        System.out.println("(2): Credit Card");
        int mode = input.nextInt();

        String paymentMethod = "";
        boolean paymentSucceed = false;
        BillingService service = null;

        if(mode == 1) {
          service = (BillingService) context.getBean("gpay-service");
          paymentMethod = "Google Pay";
          paymentSucceed = service.completePayment();
        } else if (mode == 2) {
          service = (BillingService) context.getBean("credit-card-service");
          paymentMethod = "Credit Card";
          paymentSucceed = service.completePayment();
    } else {
      System.out.println("Choose between 1 or 2");
    }

        if(paymentSucceed) {
          System.out.println("Succesfully payed with " + paymentMethod);
        }

    }
```

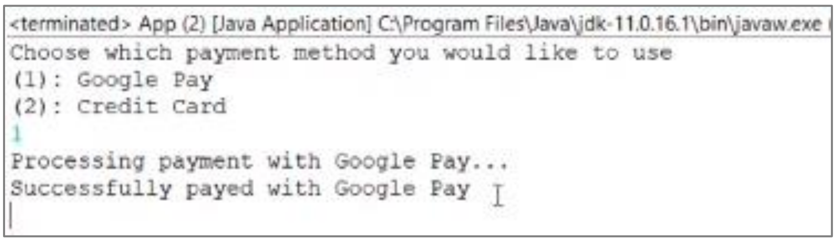
✓ Config.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd">

  <!-- Factory Bean -->
  <bean id="factory" class="lithan.training.javawebapp.PaymentProcessorFactory">
</bean>

  <!-- Google Pay -->
  <bean id="gpay-method" factory-bean="factory" factory-method="getPaymentProcessor">
    <constructor-arg name="mode" value="gpay"></constructor-arg>
  </bean>

  <bean id="gpay-service" class="lithan.training.javawebapp.BillingService">
    <property name="paymentProcessor" ref="gpay-method"></property>
  </bean>

  <!-- Credit Card -->
  <bean id="credit-card-method" factory-bean="factory" factory-
method="getPaymentProcessor">
    <constructor-arg name="mode" value="credit_card"></constructor-arg>
  </bean>

  <bean id="credit-card-service" class="lithan.training.javawebapp.BillingService">
    <property name="paymentProcessor" ref="credit-card-method"></property>
  </bean>
</beans>
```

✓ Output



```
<terminated> App (2) [Java Application] C:\Program Files\Java\jdk-11.0.16.1\bin\javaw.exe (
Choose which payment method you would like to use
(1): Google Pay
(2): Credit Card
2
Processing payment with Credit Card...
Successfully payed with Credit Card Payment
```



```
<terminated> App (2) [Java Application] C:\Program Files\Java\jdk-11.0.16.1\bin\javaw.exe (
Choose which payment method you would like to use
(1): Google Pay
(2): Credit Card
1
Processing payment with Google Pay...
Successfully payed with Google Pay
```

- ✓ **Source Code**

Assignment 1.zip