Assignment -1

|  |  |
| --- | --- |
| **Student Name/ID Number:** | Agung Yuda Pratama / 0922-076 |
| **Academic Year:** | 2022 |
| **Unit Assessor:** | Archana Sakpal |
| **Project Title:** | Assignment 1– Inversion of Control |
| **Issue Date:** |  |
| **Submission Date:** |  |
| **Internal Verifier Name:** | Archana Sakpal |
| **Date:** |  |

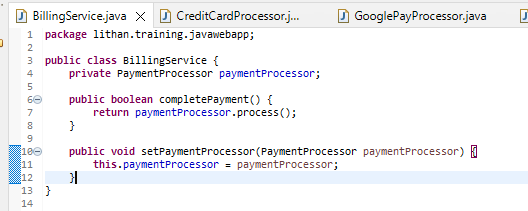
|  |
| --- |
| **Learner declaration** |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.    11/04/2023  Student signature: Date: |

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

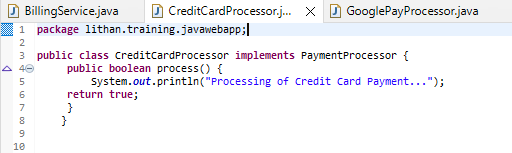
|  |  |
| --- | --- |
| **Lab Assessment Statement** | **Assignment 1 - Inversion of Control**  **You are handling the Billing module of an e-commerce application. You have developed below class.**  package lithan.training.javawebapp  public class BillingService {  private PaymentProcessor paymentProcessor;    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 mins |

**Solution:-**

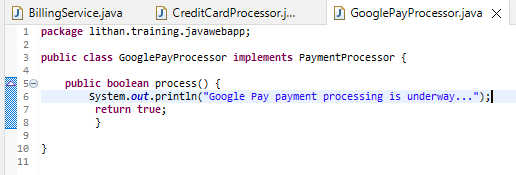
* + - 1. Source code of BillingService Class



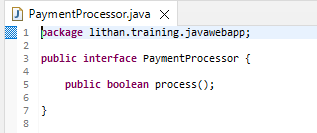
* + - 1. Source code of CreditCardProcessor Class



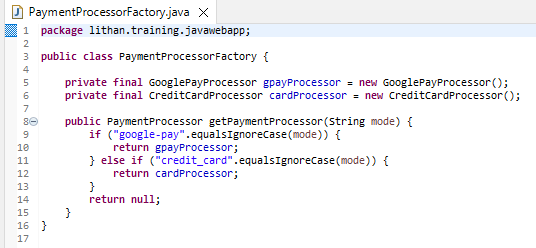
* + - 1. Source code of GooglePayProcessor Class



* + - 1. Source code of PaymentProcessor Interface



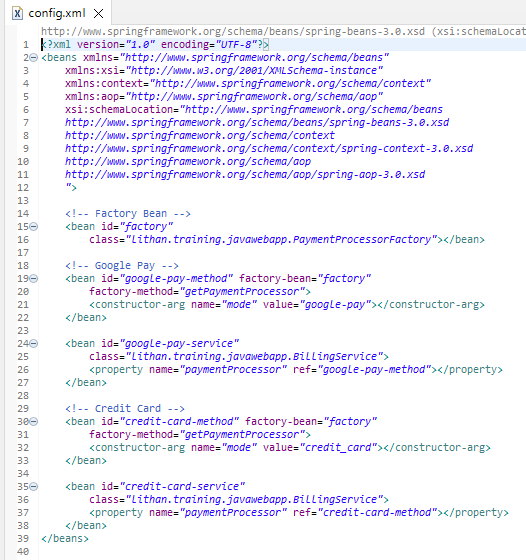
* + - 1. Source code of PaymentProcessorFactory Class



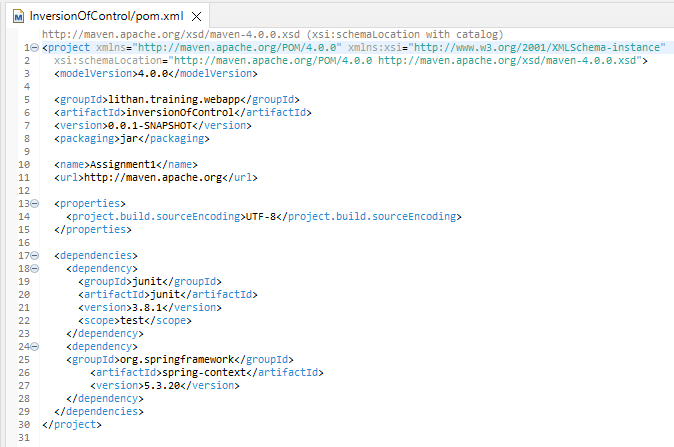
* + - 1. Source code of MyApp.java



* + - 1. Source code of config.xml file

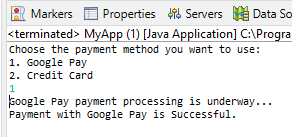


* + - 1. Source code of pom.xml file.

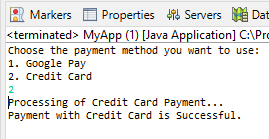


* + - 1. Screen capture of Output file.

9.1



9.2



9.3

