UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 7 Examination in Engineering: May 2023

Module Number: EE7210 **Module Name: Object Oriented Design Patterns and Principles**

[Three Hours]

[Answer **all questions**, each question carries **10** marks]

Q1 a) What is the dependency inversion principle?

[2.0 Marks]

b) Refactor the "ProductService" class to solve the dependency inversion principal violation in the code given below.

```
class ProductService {
  private final Database database;

  public ProductService() {
    this.database = new Database();
  }

  public void addProduct(Product product) {
    database.save(product);
  }

  public void updateProduct(Product product) {
    database.update(product);
  }
}

class Database {
  public void save(Product product) {
    // Save product to database
  }

  public void update(Product product) {
    // Update product in database
  }
}

class Product {
  private String name;
  private double price;
```

```
    // Getters and setters for name and price
    // Constructor, equals, and hashcode methods
}
```

[4.0 Marks]

c) A violation of Dependency inversion principle may cause to violate the open close principle. Explain how this may occur with a suitable example.

[4.0 Marks]

Q2 a) Mention 3 benefits of using GoF design patterns when writing a software.

[3.0 Marks]

b) Answer the (i) and (ii) based on the following code.

```
class PaymentProcessor {

    private PaymentGateway paymentGateway;


    public PaymentProcessor() {
        this.paymentGateway = new PaymentGateway();
    }


    public void processPayment(String paymentMethod, double amount) {
        if (paymentMethod.equals("credit_card")) {
            paymentGateway.processCreditCardPayment(amount);
        } else if (paymentMethod.equals("paypal")) {
            paymentGateway.processPaypalPayment(amount);
        } else {
            throw new IllegalArgumentException("Invalid payment method");
        }
    }
}

class PaymentGateway {

    public void processCreditCardPayment(double amount) {
        // Process credit card payment
    }

    public void processPaypalPayment(double amount) {
        // Process PayPal payment
    }
}
```

(i) List down the problems in the code in terms of maintainability.

[2.0 Marks]

(ii) Refactor the above code by using a suitable design pattern(s).

[5.0 Marks]

Q3 a) Briefly explain the "Indirection" in GRASP (General Responsibility Assignment Software Patterns).

[2.0 Marks]

b) Propose a Gang of Four design pattern that is similar to the "Indirection" in GRASP and explain why they are similar.

[3.0 Marks]

c) **"Duplicate code is an enemy of the quality."** What are the consequences of duplicate code?

[5.0 Marks]

Q4 a) Why mocking is important for unit testing?

[2.0 Marks]

b) Answer the (i) and (ii) based on the following code.

```
public class ShoppingCart {
  private List<Product> products = new ArrayList<>();
  private double totalPrice = 0;


  public void addProduct(Product product) {
    products.add(product);
    totalPrice += product.getPrice();
  }


  public void removeProduct(Product product) {
    products.remove(product);
    totalPrice -= product.getPrice();
  }


  public double getTotalPrice() {
    return totalPrice;
  }
```

```
    public int getProductCount() {
        return products.size();
    }

    public void checkout(PaymentGateway gateway) {
        gateway.processPayment(totalPrice);
    }
}
```

(i) Write a unit test in Java to cover the scenario "The checkout should process payment when a payment gateway is provided".

[4.0 Marks]

(ii) List down 3 other unit test scenarios that can be implemented.

[4.0 Marks]

Q5 a) What are the benefits of using a static code analysis tool?

[2.0 Marks]

b) "Test Driven Development (TDD) saves the development time and the maintenance time in a project" Justify this statement.

[4.0 Marks]

c) What is refactoring, and how does it differ from rewriting code?

[4.0 Marks]