

The Open University of Sri Lanka  
Faculty of Engineering Technology  
Department of Electrical & Computer Engineering



|                         |                                                                                           |
|-------------------------|-------------------------------------------------------------------------------------------|
| Study Programme         | : Bachelor of Science Honours in Engineering/<br>Bachelor of Software Engineering Honours |
| Name of the Examination | : Final Examination                                                                       |
| Course Code and Title   | : EEI4362/EEX4362 Object Oriented Design                                                  |
| Academic Year           | : 2024/25                                                                                 |
| Date                    | : 10 <sup>th</sup> February 2026                                                          |
| Time                    | : 0930-1230h                                                                              |
| Duration                | : 3 hours                                                                                 |

### General instructions

1. Read all instructions carefully before answering the questions.
2. This question paper consists of **Four (4)** questions in **Four (4)** pages.
3. Answer **all** questions.
4. All questions carry equal marks.
5. Answer for each question should commence from a new page.
6. This is a Closed Book Test (CBT).

Consider the following scenario to answer **Question 01 & 02**

Hela Games Inc. software company in Sri Lanka plans to develop a Chess Game application. The game should support two players playing against each other. The chessboard consists of 64 squares arranged in an 8×8 grid. Each player has a set of chess pieces such as King, Queen, Rook, Bishop, Knight, and Pawn. Each piece has its own movement rules. The game should keep track of the current player, validate moves, and determine game states such as check and checkmate. (Check and checkmate are two different states of the game.)

The **ChessGame** class is responsible for controlling the overall flow of the game, including starting the game, switching turns between players, and checking the game status such as check or checkmate. The ChessGame is associated with two(2) **Player** objects, representing the two players participating in the game. Each Player has attributes such as name and color and is responsible for making moves during the game. The **Board** represents the chessboard and consists of 64 Square objects. Each **Square** has a position and may contain one Piece at a given time. The Square cannot exist without the Board. The Board also has an association with the ChessGame. The **Piece** is an abstract that defines common attributes such as color and position, along with common behaviour like move and isValidMove. Specific chess pieces such as **King, Queen, Rook, Bishop, Knight, and Pawn** are children of the Piece. These children override the move() method to implement their own movement rules.

### Question 01

- a. As an object-oriented designer, identify the relevant classes, their attributes, methods, and relationships, including a minimum of five(5) valid assumptions, and design a suitable **class diagram** for the Chess Game App. (20 marks)
- b. Write a code snippet in Java to create a **persistent method** named `savePlayer()` using Hibernate session and transaction to insert data for the persistent class **Player** from the above scenario. (05 marks)

### Question 02

- a. Design a sequence diagram for the scenario where **Player1** sends a request to start the game to the **ChessGame** object. The ChessGame object then tries to connect a **Player2** by sending a request. Player2 can either approve or reject the request sent by Player1. If Player2 rejects the request, the ChessGame object repeats the process until a Player2 approves the request. Once Player2 approves the request, the ChessGame object sends a request called "createNewBoard" to the Board object. The Board object then sends the boardUI to both Player1 and Player2. After that, the ChessGame object sends a request called "createPieces" to the Piece object. The Piece object sends the piecesUI to both Player1 and Player2, making the chessboard and pieces visible to both players. (10 marks)
- b. Design an activity diagram for the scenario where the game begins when Player1 makes a move. Then the system checks whether Player2's king is in the state "check" (*a situation where a player's king is under threat*). (10 marks)

If the king is in check state, Player2 receives a message "king is under threat" and Player2 must respond by attempting a defensive move. Once Player2 makes the defensive move. The system validates whether the move is valid and whether it successfully removes the check. If the move is valid and removes the check, the check is considered defended and then the turn passes back to Player1. However, if the move is invalid and does not remove the check, Player2 must attempt another defensive move. Before allowing Player2 to attempt again, the system checks whether any valid defensive move exists. If no valid defensive move is available, the game declares checkmate and the game ends. If Player1's initial move does not place the king in check, the game continues with normal play. Finally, the turn returns to Player1.

- c. State five(5) advantages of using UML diagrams when designing an application for the above scenario? (05 marks)

## Question 03

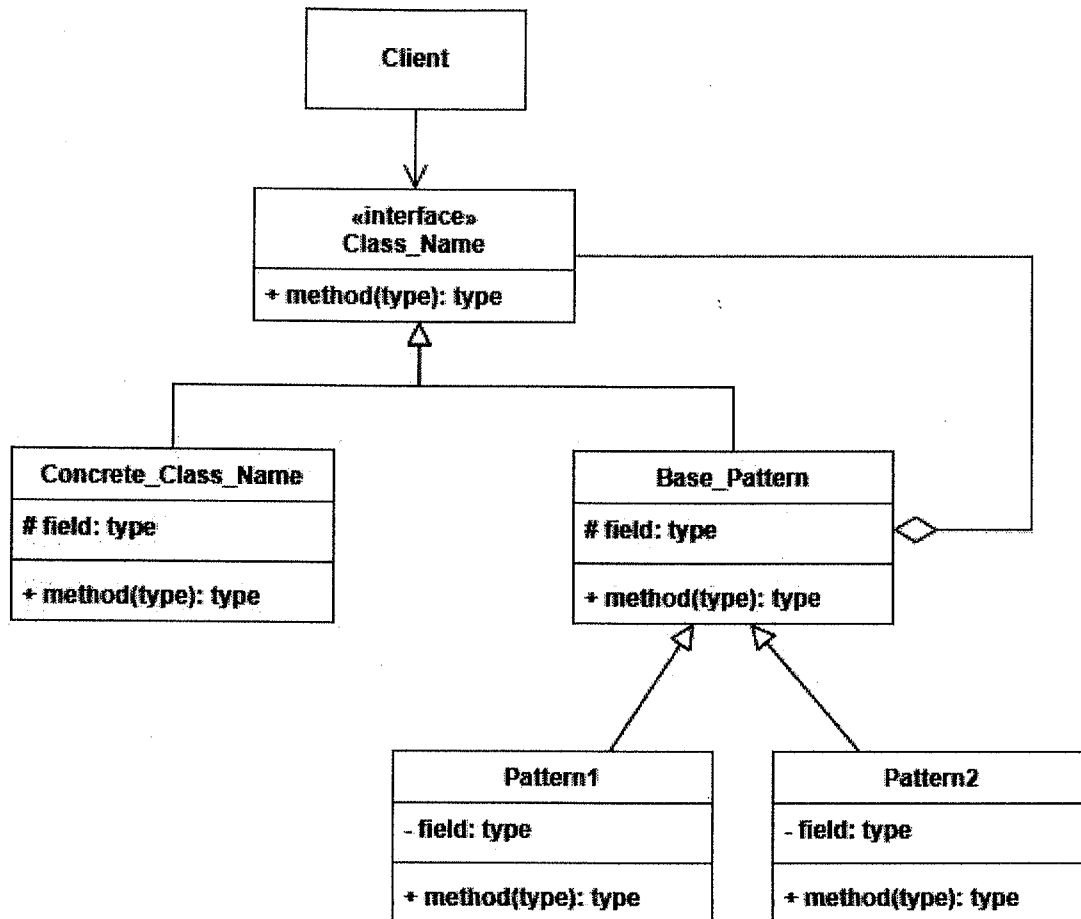


Figure 01

Answer the questions below based on **Figure 01**.

- Identify the design pattern illustrated and state the category to which it belongs. (04 marks)
- Briefly explain the **Problem**, **Solution**, and **Consequences** of the identified design pattern in part a. (06 marks)
- Discuss four(4) advantages of using the illustrated design pattern. (04 marks)
- Identify the participants of the design pattern and briefly explain the role of each participant. (08 marks)
- Briefly explain the three(3) relationships shown in the illustrated design pattern. (03 marks)

#### Question 04

- a. Briefly explain how you can create your own Thread object in two(2) (04 marks) different ways.
- b. Answer the questions based on the given Java code.

```
class TestSleepMethod1 extends Thread {  
    public void run() {  
        for (int i = 1; i < 5; i++) {  
            try {  
                Thread.sleep(1000);  
            } catch (InterruptedException e) {  
                System.out.println(e);  
            }  
            System.out.println(Thread.currentThread().getName() + " - " + i);  
        }  
    }  
}
```

```
public class ThreadDemo {  
    public static void main(String[] args) {  
        TestSleepMethod1 t1 = new TestSleepMethod1();  
        TestSleepMethod1 t2 = new TestSleepMethod1();  
  
        t1.start();  
        t2.start();  
    }  
}
```

- I. What is the output of the above Java code if both threads start at the exact same time? (04 marks)
- II. What is the output when invoking the run( ) method on above threads instead of start()? Briefly explain the difference. (06 marks)
- III. Briefly explain the purpose of the join( ) method in Java threads. (05 marks)
- IV. Write code to make the above threads run sequentially using start() and join(). Handle exceptions while using join() (06 marks)

*\*All Rights Reserved\**