00469

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

Study Programme	: Bachelor of Software Engineering Honours : Bachelor of Technology Honours in Engineering
Name of the Examination	: Final Examination
Course Code and Title	: **EEI/EEX4362 Object Oriented Design**
Academic Year	: 2022/2023
Date	: 22nd October 2023
Time	: 0930 -1230 hrs
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. Answer for each question should commence from a new page.
5. This is a Closed Book Test (CBT).
6. Answers should be in clear handwriting.
7. Do not use red colour pen.

Question 01 [40 Marks]

Consider the following scenario.

A poker game is a card game that combines elements of strategy, skill, and chance. Players use a standard deck of 52 cards and aim to win chips or money from their opponents by having the best hand or by bluffing to convince other players to fold.

You are tasked with designing a software system for few functionalities of a poker game. The game should support multiple players, each with their own hand of cards, and should be able to handle different variations of poker games, such as Texas Hold'em or Five Card Draw. Players should be able to place bets, receive cards, and determine the winner based on the rules of the specific poker game being played.

a) Draw a use case diagram to illustrate interactions between a **Player** representing a player participating in the game, and a **Dealer**, representing the dealer who manages the game. Include the following in your use case.

The dealer initiates a new poker game, involving multiple players. The dealer starts each round by shuffling the deck of cards to randomize their order. Then, they distribute a specific number of cards to each player, typically face down. This act is called as "dealing cards". In some games, the dealer also places community cards on the table that are shared by all players. These community cards are typically dealt face up in the centre of the table and are part of the public information that all players can use to form their best possible hand. The number of community cards and the specific rules for their use vary from game to game. The chips or money collected from bets are placed in a central area, separate from each player's stack. This centralized pool is known as the "pot". The dealer manages the bets during the game, including collecting chips from losing players, awarding chips to the winner, managing the pot, shuffling etc.

Players can place bets during their turn. Additionally, players can Fold, Check, Call, Raise and All-In while playing a game. These are several options players have when it is their turn to act. For an example if it is All-In, player bets all the chips, they have left. The system determines the winner based on the poker rules. Include any assumptions made while drawing the diagram. ***(08 Marks)***

b) Design a class diagram for the poker game software system that includes classes. Each player should have a name, a current hand of cards (fixed number of cards), and the amount of money they have for betting. Include methods for adding cards to a player's hand and updating their balance. The card represents a playing card, which has a suit (hearts, diamonds, clubs, or spades) and a rank (2 through 10, Jack, Queen, King, Ace). The deck should include methods for shuffling the deck and dealing cards to players or the community. The game represents the poker game itself. Include methods for starting a new game, determining the currently active player, managing bets, and ending the game to declare a winner. Community cards should represent the common cards that are placed face up on the table in games like Texas Hold'em. Include methods for adding community cards as the game progresses. Poker rules is a class that encapsulates the rules that are specific to the poker game being played and includes methods for evaluating hands and determining winners. Bets should include information about the player making the bet and the amount and any relevant information for tracking bets. The pot represents the pot of money that players are betting into. Include methods for adding and distributing money to/from the pot. Ensure that you specify relationships between the classes, such as players having hands of cards and the game using a deck and community cards. Your class diagram should depict how these classes interact to facilitate the poker game. ***(22 Marks)***

c) Create a sequence diagram that illustrates the interactions between objects for the "Start New Game" scenario, which should involve interactions between the Player, Deck, PokerGame, PokerRules, and Pot. The sequence diagram should depict the flow of actions and messages between the objects given below.

The "Player" signals the "PokerGame" to start a new game. The "PokerGame" activates and initializes the game. The "PokerGame" uses the "Deck" to shuffle the cards and deal them to the "Player". Each "Player" receives cards and updates their hands. The "PokerGame" manages the bets, and players place bets. The "PokerGame" uses the "PokerRules" to evaluate the hands of players. The "PokerGame" adds bets to the "Pot" as the game progresses. The "PokerGame" ends the game and determines the winner based on the rules from the "PokerRules". The "Pot" is distributed to the winning player(s).

(10 Marks)

Question 02 [20 Marks]

a) Give three differences between an interface and an abstract class in Java. ***(03 Marks)***

b) A novice programmer writes the following code to define an API for a payment gateway for an E-commerce business.

```
interface PaymentGateway {
   void processPayment(double amount);
   void refundTransaction(); }

abstract class AbstractPaymentGateway implements PaymentGateway {
   private String transactionId;
   public abstract String checkTransactionStatus();

public void processPayment(double amount) {
      // Common code for processing payment }

public void refundTransaction() {
      // Common code for refunding transaction }

   }
```

```
class PayPalPaymentGateway extends AbstractPaymentGateway {

public void processPayment(double amount) {
      // PayPal payment processing code    }

   public void refundTransaction() {
      System.out.println("Refund requested for " +transactionId);
        // PayPal refund transaction code      }
  }

class PaymentAPI
{
  public static void main (String args[])

    { PaymentGateway paypalGateway = new PayPalPaymentGateway();}
}
```

i. What does it signify when the transactionId field is private, and why is it considered good programming practice for such fields to be private? ***(03 Marks)***

ii. Identify and explain the type of inheritance used in this program. ***(02 Marks)***

iii. When you run this program, it will create compile time errors. Identify the reasons for those errors. ***(04 Marks)***

iv. Rewrite the code to correct the problems you have identified in (iii). ***(08 Marks)***

Question 03 [20 Marks]

a) What are the two different ways to create a thread in Java? ***(04 Marks)***

b) Explain the lifecycle of a thread using an appropriate diagram. ***(08 Marks)***

c) Imagine a banking system where multiple users are performing transactions on their bank accounts concurrently using multithreading. How would you ensure that account balances are updated correctly and consistently? ***(04 Marks)***

d) Consider the following program

```
class NumberSystem extends Thread{
 public void run(){
  for(int i=1;i<=5;i++){
    try{Thread.sleep(500);}

    catch(InterruptedException e)
        {System.out.println(e),}
    System.out.print(i);
   }
 }
 public static void main(String args[]){
  NumberSystem t1=new NumberSystem ();
  NumberSystem t2=new NumberSystem ();
  t1.start();
  t2.start();
 } }
```

i. Identify the output of the above program. ***(02 Marks)***

ii. What is the role of the sleep() method in the program? ***(02 Marks)***

Question 04 [20 Marks]

a) What is the use of design patterns in software design and programming? ***(02 Marks)***

b) What are the four main essential elements of a design pattern? ***(04 Marks)***

c) Answer the questions based on the given scenario

In a restaurant application, a menu can be composed of individual items or submenus, each of which can further contain items or submenus.

i. Which is the appropriate design pattern to represent the menu tree structure? ***(02 Marks)***

ii. Discuss any two advantages of using the selected design pattern. ***(04 Marks)***

iii. Explain the participants of the selected design pattern. ***(08 Marks)***