

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)
ORGANISATION OF ISLAMIC COOPERATION (OIC)
Department of Computer Science and Engineering (CSE)

MID TERM EXAMINATION**SUMMER SEMESTER, 2019-2020****DURATION: 1 HOUR 30 MINUTES****FULL MARKS: 75****CSE4851: Design Pattern**

Write your **Name, Student-ID, and Course Code** on the top of the first page.

Put a serial number on the Top-right corner of each page

There are **3 (Three)** questions. Answer **All** of them. Figures in the right margin indicate marks.

1. (a) What is Design Pattern? Describe two key advantages of applying design patterns. Name the type of the patterns. 5
- (b) For each part, write down the name of the design pattern or principle that would be most useful for addressing the situation described. 6
 - i. You are building a system that relies on a complex algorithm, and that algorithm may be changed often due to marketing pressures. What pattern would best support this?
 - ii. A pizza factory produces pizzas with various toppings. There are 20 different toppings and a customer may order any combination of toppings. Assume that each of pizza bread and each topping will be represented by a different class.
 - iii. We are building a cricket app that notifies viewers about the information such as current score, run rate etc. Suppose we have made two display elements CurrentScoreDisplay and AverageScoreDisplay. CricketData has all the data (runs, bowls etc.) and whenever data changes the display elements are notified with new data and they display the latest data accordingly.
- (c) Explain a scenario where strategy pattern can be used. Write the corresponding code for that scenario. Also, draw the UML diagram for that scenario. 4+5+5
2. (a) Briefly, explain the purpose of the Decorator pattern. List three distinct advantages of factory methods over constructor. 5
- (b) Draw a UML diagram for Mediator pattern between web services and web clients. As web services, the Ebay auction house and Amazon are available. Plan functions to search for an item with a textual description, and to buy an item from the service that gives you the best price. 10
- (c) Identify a pattern which can Define a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically. Briefly explain that pattern. Also discuss the advantages and disadvantages of that pattern. 10
3. (a) We have used the term “program to an interface, not to an implementation”. Explain a pattern satisfying the statement with real world scenario. 5
- (b) Briefly discuss the usage of composition over inheritance. 5
- (c) Consider the following code- 5+5+5

```

public class Rental {
    private Movie _movie;
    private int _daysRented;

    public Rental(Movie movie, int daysRented) {
        _movie = movie;
        _daysRented = daysRented;
    }
    public int getDaysRented() { return _daysRented; }
    public Movie getMovie() { return _movie; }

    public double amountFor() {
        double thisAmount = 0;
        //determine amounts for each line
        switch (getMovie().getPriceCode()) {
            case Movie.REGULAR:
                thisAmount += 2;
                if (getDaysRented() > 2)
                    thisAmount += (getDaysRented() - 2) * 15;
                break;
            case Movie.NEW_RELEASE:
                thisAmount += getDaysRented() * 3;
                break;
            case Movie.CHILDRENS:
                thisAmount += 1.5;
                if (getDaysRented() > 3)
                    thisAmount += (getDaysRented() - 3) * 1.5;
                break;
        }
        return thisAmount;
    }
}

public class Movie {
    public static final int CHILDRENS = 2;
    public static final int REGULAR = 0;
    public static final int NEW_RELEASE = 1;

    private String _title ;
    private int _priceCode;

    public Movie(String title , int priceCode) {
        _title = title ;
        _priceCode = priceCode;
    }

    public int getPriceCode() { return _priceCode; }
    public void setPriceCode(int arg) { _priceCode = arg; }
    public String getTitle () { return _title ; }
}

```

- i) Briefly explain the terms code refactoring and code smell.
- ii) Identify two code smells which occur in the code.
- iii) Refactor the code removing the smells.