B.Sc. Engg. CSE 8th Semester **14 February, 2022 (Morning)**

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION **SUMMER SEMESTER, 2020-2021**

DURATION: 1 Hour 30 Minutes **FULL MARKS: 75**

CSE 4851: Design Pattern

Programmable calculators are not allowed. Do not write anything on the question paper.

Answer **all 3 (three)** questions. Marks of each question and corresponding CO and PO are written in the right margin with brackets.

---

1. a) What are the main components of OOP? "Classes should be open for extension and closed to modification"- Explain the statement with proper example. 5 (CO1) (PO1)

   b) For each of the following part, write down the name of the design pattern or principle that would be most useful for addressing the situation described: 3x2 (CO4) (PO2)

      i. You are building a system that relies on a complex algorithm, and that algorithm may be changed often due to marketing pressures.

      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, balls, 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 Adapter pattern can be used. Write the corresponding code for that scenario. Also, draw the UML diagram for that scenario. 14 (CO4) (PO2)

2. a) What are the SOLID principles? Explain with example pseudocode. 7 (CO1) (PO1)

   b) Mention the purpose of following patterns: 2x3 (CO3) (PO1)

      i. Proxy

      ii. Memento

   c) Which design pattern decouples an abstraction from implementations? Explain a scenario satisfying that pattern. Draw a UML diagram for that scenario. 2+5+5 (CO4) (PO2)

3. a) "Program to an interface, not to an implementation" - Explain a pattern satisfying the statement with real world scenario. 2+5 (CO4) (PO2)

   b) Which design pattern uses composition to extend the capabilities of an object at runtime? Explain a scenario satisfying that pattern. Draw a UML diagram for that scenario. 2+5+5 (CO4) (PO2)

c) Differentiate between: 3x2 (CO3) (PO2)

i. Method overloading and overriding

ii. Factory and Abstract Factory, and

iii. Coupling and Cohesion