40

B.Sc. Engg. CSE 8th Semester — 20 February 2023 (Morning)

ISLAMIC UNIVERSITY OF TECHNOLOGY (IUT)

ORGANISATION OF ISLAMIC COOPERATION (OIC)

Department of Computer Science and Engineering (CSE)

MID SEMESTER EXAMINATION — **SUMMER SEMESTER, 2021-2022**

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. Figures in the right margin indicate full marks of questions whereas corresponding CO and PO are written within parentheses.

1. a) What are the main components of OOP? Explain "Dependency Inversion Principle" with a code example. 2+5 (CO1) (PO1)

   b) For each part, write down the name of the design pattern or principle that would be most useful for addressing the situation described. Explain the reason for choosing the pattern or principle. 6 (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. 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 pizza bread and each topping will be represented by a different class.

   c) Explain a scenario where Template Pattern can be used. Write the corresponding code for that scenario. Also, draw the UML diagram for that scenario. 12 (CO4) (PO1)

2. a) Briefly explain the purpose of the Decorator Pattern. List three distinct advantages of factory methods over constructor. 5 (CO3) (PO1)

   b) Imagine a situation where you have a software called `KnifeStore` that implements an online store that sells knives. The software produces `Steakknives` and `Chefsknives`. The methods of sharpening, polishing, and packaging would be defined in the `orderKnife` method. However, the responsibility of creating the product will be delegated to another class; a `KnifeFactory`. Subclass of the `KnifeFactory` called `BudgetKnifeFactory` would make `BudgetChefsKnife` and `BudgetSteakKnife` product objects. Subclass of the `KnifeFactory` called `BasicKnifeFactory` would make `BasicChefsKnife` and `BasicSteakKnife` product objects. 20 (CO5) (PO2)

      Write the implementation of above scenario using appropriate pattern and draw the UML diagram of your implementation.

3. a) Which design pattern works as a bridge between two incompatible interfaces? Explain the intent and motivation of this pattern. 7 (CO3) (PO1)

   b) Which design pattern uses composition to extend the capabilities of an object at runtime? Explain a scenario satisfying that pattern. Draw an UML diagram for that scenario. 12 (CO3) (PO1)

   c) Differentiate between the following patterns: 6 (CO3) (PO1)

      i. Prototype and Singleton

      ii. Strategy and Decorator

      iii. Coupling and Cohesion