

**University of Central Punjab**

**Faculty of Information Technology**

**Department of Software Engineering**

**Software Design and Architecture (S23)**

**Assignment No. 03**

Instructor: Dr Hafiz Mahfooz Ul Haquie

|  |  |
| --- | --- |
| **Student 01 Name:** | Waqar Asghar |
| **Roll Number:** | L1S21BSSE0119 |

**Important Instruction**

1. It is a group assignment. Each group shall consist of two members.
2. Active participation of each group member is advised.
3. Assignment has one question which has three parts.
4. The assignment is based on the topic/s we have discussed during class lecture.
5. Spend some good time making yourself comfortable with the concepts to be applied in the assignment.
6. Copying from any source shall result in zero marks.
7. Please take into consideration the dead line for the submission.

**Good Luck**

**Total Marks: 30**

**Question No. 01** **[30 Marks]**

**Please read the following scenario and then answer part A, B and C.**

**Scenario:**

Suppose you are working in a software company. The company has a got a project to design and implement an online food ordering system for a restaurant chain. The system should allow customers to place orders for various types of food items, such as pizza, burgers, and salads. Each type of food item has its own *preparation* and *cooking* methods.

Your task is to design a system that encapsulates the food item creation process while allowing subclasses to define the specific food items they want to create.

1. **Your task is to propose the GoF design pattern name that will be appropriate for the above scenario. You must provide justification.**  **[05 Marks]**

The appropriate GoF design pattern for the given scenario would be the “Factory Method pattern.”

**Justification:**

The Factory Method pattern is a design approach that allows us to create different types of food items in an online food ordering system. It encapsulates the process of creating food items and lets subclasses define the specific food items they want to create. In this pattern, we have a factory class or interface that declares a factory method responsible for creating objects. Concrete factories, which are subclasses of the factory class, implement this factory method to create specific food items like pizza, burgers, or salads. This pattern provides benefits such as encapsulation, making it easy to add new food items without modifying existing code, customization of creation logic for each food item, and reusability of common creation code. Overall, the Factory Method pattern helps us create and manage various food items in a flexible and extensible way for the online food ordering system.

1. **Create UML Diagram to reflect the class design for the above scenario.** **[10 Marks]**

1. **Write code in C++/Java that must be 100% synched with the class diagram created in part B   [15 Marks]**