|  |  |  |
| --- | --- | --- |
| **Course** | **:** | Design Patterns |
| **Teacher** | **:** | Sir Arsalan Qureshi |
| **Assignment No** | **:** | 2 |
|  |  |  |
| **Student** | **:** | Farrukh Afzaal Kurd |
| **Student ID** | **:** | FA17 – MSCS – 0022 |

**Abstract Diagrams – Design Patterns**

1. **Singleton Pattern - Abstract Diagram**

return uniqueinstance

**Singleton**

static instance()

SingletonOperation()

GetSingletonData()

static uniqueinstance

singletonData

1. **Factory Pattern - Abstract Diagram**

**Product**

**ConcreteProduct**

**Creator**

FactoryMethod()

AnOperation()

Product = FactoryMethod()

return new ConcreteProduct

**ConcreteCreator**

FactoryMethod()

1. **Adapter Pattern - Abstract Diagram**

***Class Adapter - Abstract Diagram***

implementation

**Client**

**Target**

Request()

SpecificRequest()

**Adapter**

Request()

**Adaptee**

SpecificRequest()

***Object Adapter - Abstract Diagram***

adaptee

**Client**

**Target**

Request()

Adaptee --> SpecificRequest()

**Adapter**

Request()

**Adaptee**

SpecificRequest()

1. **Decorator Pattern - Abstract Diagram**

**Decorator**

Operation()

component --> Operation()

Decorator :: Operation();

AddedBehavior();

**ConcreteDecoratorB**

FactoryMethod()

**ConcreteDecoratorA**

Operation()

addedState

**ConcreteComponent**

Operation()

**Component**

Operation()

component