Bahria University
Discovering Knowledge
Department of Software Engineering

SEN-221-SOFTWARE DESIGN & ARCHITECTURE
SPRING 2022

COURSE INSTRUCTOR
ENGR. MAJID KALEEM

Assignment No.	02
Assignment Title	Description of various Architectural Styles
Course Learning Outcome	CLO-03 *"Apply design models using modeling and object-oriented programming languages."*
Full Name	M Muaz Shahzad
Semester	BSE 4 B
Submission Deadline	26th April 2022

INSTRUCTIONS:

- ☐ USE ONLY TIMES NEW ROMAN SIZE 12 FONT.
- ☐ EACH HEADING (UNDERLINED, BOLD AND IN CAPITAL LETTERS) AND EXAMPLE MUST START FROM A NEW LINE.
- ☐ UPLOAD SOFTCOPY ON LMS AS A PDF FILE.
- ☐ DO NOT EDIT (THIS) ASSIGNMENT FILE GIVEN AS A PDF FILE.
- ☐ LAST PAGE OF YOUR ASSIGNMENT MUST CONTAIN SOURCES/REFERENCES (USE IEEE REFERENCING STYLE).
- ☐ NO MAKEUP ASSIGNMENTS WILL BE GIVEN & DATE WILL NOT BE EXTENDED.
- ☐ VIOLATION OF ANY OF THE INSTRUCTIONS MENTIONED HERE WILL RESULT IN MARKS DEDUCTION.

BAHRIA UNIVERSITY (KARACHI CAMPUS)

Software Design & Architecture (SEN -221)

ASSIGNMENT # 2 – Spring 202 2

Based on: CLO - 3

Submission Deadline: **26th April 22**

Class: **BSE-4B**

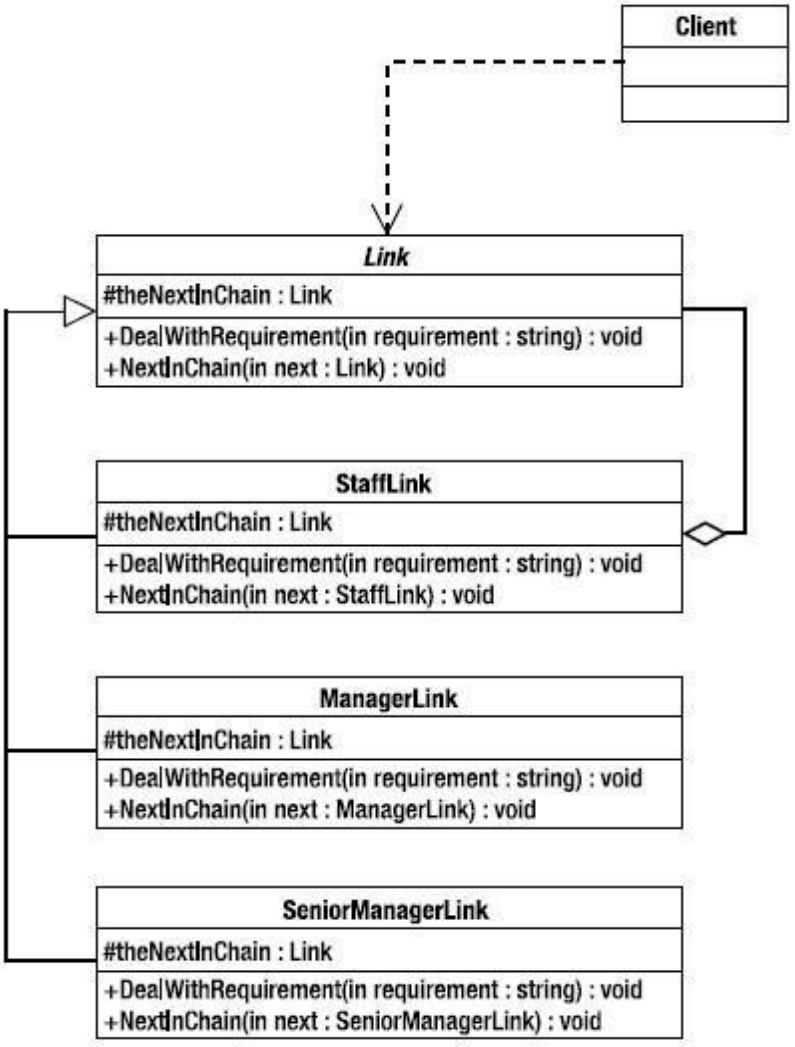
ENGR. MAJID KALEEM

Max Marks: **05**

Course Instructor:

1. Designing software applications is a serious job which requires experience and expertise. Suppose you start your professional software engineering career as a developer and you are given the following designs by your senior team members. Your task is to convert (*produce code in C Sharp*) the following designs into code. Please write the code in Visual Studio.

(a)

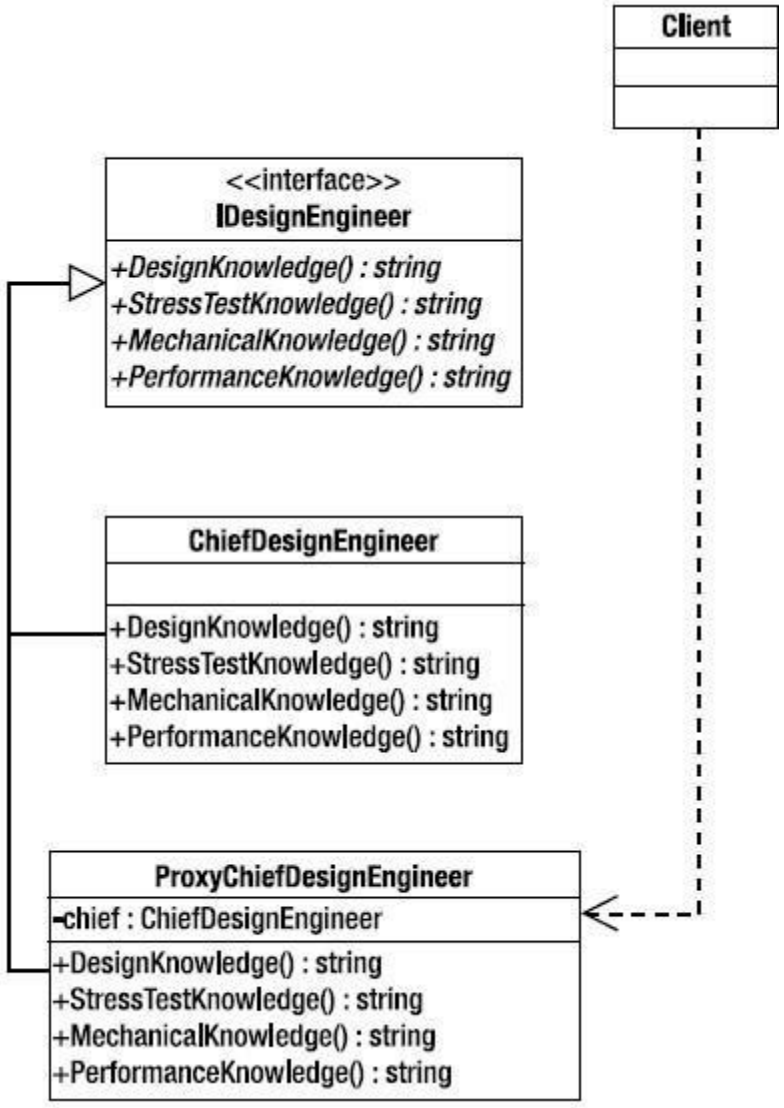


PROGRAM:

```
using System;

public class tasks
{
   class Clients
   {
      Link link;
     public void createlink ()
      {
         link = new Link();
      }
   }

   class Link
   {
      Link thenextlnchain;

      public void Dealwithrequirement(string in_requirement)
      { }
      public void Nextlnchain(Link in_next)
      { }
   }
   class StaffLink : Link
   {
      Link l1;
      public StaffLink(Link l1)
      {
         this.l1 = l1;
      }
      Link thenextlnchain;

      public override void Dealwithrequirement(string in_requirement)
      { }
      public override void Nextlnchain(StaffLink in_next)
      { }
   }

   class ManagerLink : Link
   {
      Link thenextlnchain;

      public override void Dealwithrequirement(string in_requirement)
      { }
      public override void Nextlnchain(ManagerLink in_next)
      { }
   }

   class SeniorManagerLink : Link
   {
      Link thenextlnchain;

      public override void Dealwithrequirement(string in_requirement)
```

```
        { }
        public override void NextInchain(SeniorManagerLink in_next)
        { }
    }
}
```

(b)



PROGRAM:

```
using System;

public class tasks
{

   class Client
   {
      ProxyCheifDesignerEngineer ProxyCheifDesignerEngineer;
      public void create_ProxyCheifDesignerEngineer()
      {
         ProxyCheifDesignerEngineer = new ProxyCheifDesignerEngineer();
      }
   }
   interface IdesignEngineer
   {
      public string DesignKnowledge()
      { }
      public string StressTestKnowledge()
      { }
      public string MechanicalKnowledge()
      { }
      public string PerformanceKnowledge()
      { }
   }


   class CheifDesigner : IdesignEngineer
   {


      public override string DesignKnowledge()
      { }
      public override string StressTestKnowledge()
      { }
      public override string MechanicalKnowledge()
      { }
      public override string PerformanceKnowledge()
      { }
   }

   class ProxyCheifDesignerEngineer : IdesignEngineer
   {
      private CheifDesigner Cheif;

      public override string DesignKnowledge()
      { }
      public override string StressTestKnowledge()
      { }
```

```
        public override string MechanicalKnowledge()
        { }
        public override string PerformanceKnowledge()
        { }
    }


}
```

G☺☹d Luck!