

*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 objectoriented programming languages.”* |
| Full Name | Muhammad Junaid Saleem Qadri |
| Enrollment Number | 02-131202-057 |
| Semester | BSE 4 B |
| Submission Deadline | 26th April 2022 |

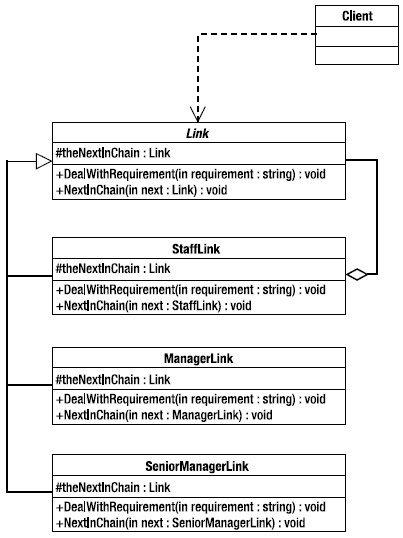
**INSTRUCTIONS:**

* 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.

Text

Description automatically generated

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)

**SOLUTION:**

class Clients

{

Link link;

public void createlink ()

{

link = new Link();

}

}

class Link

{

protected Link theNextInChain;

public Link()

{ }

public void DealWithRequirement(string requirement)

{ }

public void NextInChain(Link next)

{ }

}

class StaffLink : Link

{

Link obj;

protected Link theNextInChain;

public StaffLink(Link obj)

{

this.obj = obj;

}

public override void DealWithRequirement(string requirement)

{ }

public override void NextInChain(StaffLink next)

{ }

}

class ManagerLink : Link

{

Link theNextInChain;

public override void DealWithRequirement(string requirement)

{ }

public override void NextInChain(ManagerLink next)

{ }

}

class SeniorManagerLink : Link

{

protected Link theNextInChain;

public override void DealWithRequirement(string requirement)

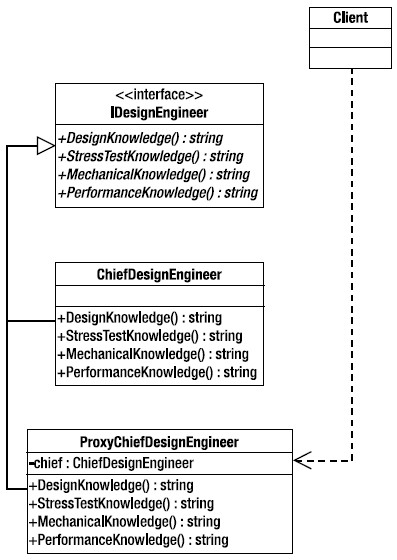
{ }

public override void NextInChain(SeniorManagerLink next)

{ }

}

(b)



**SOLUTION:**

interface IDesignEngineer

{

public string DesignKnowledge();

public string StressTestKnowledge();

public string MechanicalKnowledge();

public string PerformanceKnowledge();

}

class ChiefDesignEngineer : IDesignEngineer

{

public ChiefDesignEngineer()

{ }

public override string DesignKnowledge()

{ }

public override string StressTestKnowledge()

{ }

public override string MechanicalKnowledge()

{ }

public override string PerformanceKnowledge()

{ }

}

class ProxyCheifDesignEngineer : IDesignEngineer

{

private CheifDesignEngineer Cheif;

public ProxyChiefDesignEngineer()

{ }

public override string DesignKnowledge()

{ }

public override string StressTestKnowledge()

{ }

public override string MechanicalKnowledge()

{ }

public override string PerformanceKnowledge()

{ }

}

class Client

{

ProxyCheifDesignerEngineer proxyEng;

public void create\_ProxyCheifDesignerEngineer()

{

proxyEng = new ProxyCheifDesignerEngineer();

}

}

Gd Luck!