

Structured
Behavioral
Encapsulation

Software Engineering

Course Code: SWE-321 Term Test # 1
Total Marks: 20 Date: 21st March, 2019

Creation

1. What are the classifications of design patterns? Name a pattern in each class. 2
2. Write the SOLID principal and explain each properties with example. 4
3. What are the basic features of OOP? Which one is the most important and why? 3
4. What is the motivation behind Command pattern? Draw the UML or class diagram for command pattern. Identify the participants and their roles from the diagram. 6
5. What is Refactoring and Code smell? 2
6. Explain what the team Feature Envy means, giving a short code or diagram example to illustrate the remedies. 3

Software Engineering

Course Code: SWE-321 Term Test # 2

Total Marks: 20 Date: 6th May, 2019

memento

1. Suppose you are developing a software which will create a class that will contain two double type fields and you will run some mathematical operations on it. Your software will provide users with the undo operation. If the results after some operations are not satisfied to a user, the user can call the undo operation which will restore the state of the object to the last saved point. 10

The application also includes a save point mechanism which is used by the user to save the state of the object. You will also provide a variety of undo operations. A simple undo would restore the object state to the previous save point. An undo with the specified save point will restore that particular state of the object and undo all will delete all the saved state of the object and restore the object in the initialized state, when the object was created.

Design the above mentioned system with appropriate design pattern. Explain the participants. Draw the class diagram and write the code supporting your design.

2. Briefly describe the 3 principal layers.
3. What is service layer? How it works?

5

5

Group A
[Answer all the questions]

5x2=10

1. Answer any FIVE
- What are the classifications of design patterns? Name two patterns in each class. ✓
 - Describe the purpose of using Factory method pattern.
 - Define Code smell and Refactoring. ✓
 - What are the characteristics of Enterprise application? ✓
 - Explain Pull up field and Collapse Hierarchy. ✓
 - Explain SOLID principals.
 - Describe the purpose of using Singleton Pattern.

2. Answer any TWO

- Explain what the term Feature Envy means, giving a short code or diagram example to illustrate the remedies. *Extract class leads more methods* **2x5=10**
- "Strategy changes the guts of an object whereas Decorator changes the skins of an object"—explain this statement with appropriate reasoning. **5**
- Which design pattern provides us a way to reduce memory requirement? Explain how this memory optimization actually works? What other design pattern automatically comes with this pattern implementation? **5**
- Which patterns seem to be similar to proxy pattern? Explain why they are similar. What are the dissimilarities that made these patterns exist separately? **5**

3. Answer any TWO

- Suppose we are creating a very simple file system. We have to maintain the data of the files and folders. Files are the basic element that can be saved in the file system. A folder contains several files in it. A folder may also contain one or more folder in it. A file has the following information: file_name, file_extention, file_size, location. A folder also has the following attributes: creation_time, folder_name, location. A folder must maintain list of folders and files in it. A folder should have the functionality to add and remove file or folder. *composite* **2x15=30**

Design the above mentioned system with appropriate design pattern. Draw the class diagram and write the code to support your design.

- Suppose you are making a text data reader system. This system intended to read large size of data such as 8GB data file. You want your system to read all the data at a time and handle with optimized memory usage. **15**

Design the above mentioned system with appropriate design pattern. Draw the class diagram and write the code to support your design.

- What are the four essential elements of Design Pattern?
- Describe the four ways a Design Pattern solves design related problems.
- What do you understand by "Finding Appropriate Objects"? Can you mention a method or two about finding appropriate objects within an environment? **5+3+5**

Group B
[Answer all the questions]

4. Answer any FIVE

- Can we replace interface with abstract class? Justify your answer.
- Write the patterns for View pattern and Input controller pattern.
- What does Delegation and Object Composition means?
- What is remote facade?
- Describe the motivation of Observer pattern.
- Write down two example of Enterprise application.
- Describe Application controller of Web presentation pattern.

5x2=10

UML Decorator

5. Answer any Two

- What are the patterns that seems to be similar to Composite pattern? Explain why they are similar, what are the dissimilarities that made these patterns exist separately?
- Name two different patterns that are used to eliminate if else statement in code? Give an example each of these two patterns explaining how these patterns help eliminating if else statement.
- What does Object inheritance and Object composition mean? What are the advantages and disadvantages of these two approaches while trying to achieve reusability?
- What is the motivation behind Memento pattern? Draw the UML or class diagram for memento pattern. Identify the participants and their roles from the diagram.

2x5=10

5

5

undo-redo

6. Answer any TWO

- What is the motivation of Decorator pattern? What are the participant object of decorator pattern? Briefly explain the role of each participant object. Show an example code that demonstrate the use of decorator pattern.
- We are supposed to create a computer game that has several scenarios. African scenario, Asian scenario, American Scenario. Each scenario has the following components, a terrain, 5 trees of same kind and 2 animals of same kind. For each scenario, we have different kind of terrain, tree and animal. To draw the scenario, we just need to draw the components of a scenario.

2x15=30

4+5+6

15

abstract factory

Design the above mentioned system with appropriate design pattern. Draw the class diagram and write the code to support your design.

- A software developer, Max, has worked on an e-commerce website. The website allows users to shop and pay online. The site is integrated with a 3rd party payment gateway, through which users can pay their bills using their credit card. Everything was going well, until his manager called him for a change in the project. The manager told him that they are planning to change the payment gateway vendor, and he has to implement that in the code.

15

The problem that arises here is that the site is attached to the Xpay payment gateway which takes an Xpay type of object. Thenew vendor, PayD, only allows the PayD type of objects to allow the process. Max doesn't want to change the whole set of 100 of classes which have reference to an object of type XPay. This also raises the risk on the project, which is already running on the production.

adapter

- Decor
- Facade
- Adapter
- Composite
- Proxy
- Fly weight
- Adapter
- Composite
- Command
- Iterator
- Strategy
- State
- Memento

Group A
[Answer all the questions]

1) What is the motivation of Template pattern? What are the participant object of template pattern? Briefly explain the role of each participant object. Show an UML diagram and example code that demonstrate the use of template pattern. 6

2) Consider a class TCPConnection that represents a network connection. A TCPConnection object can be in one of several different states: Established, Listening, Closed. When a TCPConnection object receives requests from other objects, it responds differently depending on its current state. For example, the effect of an Open request depends on whether the connection is in its Closed state or its Established state. 6

State Pattern

Based on this behavior, state a solution applying appropriate design pattern. Mention the participants for the pattern and provide UML diagram.

3) Explain what the term Oddball Solution means, giving a short code or diagram example to illustrate the remedies. 3

Group B
[Answer all the questions]

4) i) "Talk to the interface, not to the concrete class" --- Explain the statement with appropriate reasoning. 4
ii) What does Delegation and Object Composition means?

5) Name two different patterns that are used to eliminate if else statement in code? Give an example each of these two patterns explaining how these patterns help eliminating if else statement. 4

6) Imagine you are an executive chef who oversees a large chain of restaurants that serves pasta. In order to ensure that dishes at all the restaurant locations are consistent, instructions are provided for making popular dishes. The two most popular dishes are spaghetti with tomato sauce and meatballs, and penne noodles with alfredo sauce and chicken. 7

Both dishes require that you boil water, cook the pasta, add the sauce, add the protein, and garnish the plate. Some of these steps have different implementation depending on which dish is being prepared – each dish has a different sauce, protein, and garnish. Other steps, however, would be the same, such as boiling the water.

Template Pattern

Design the above mentioned system with appropriate design pattern. Draw the class diagram and write the code to support your design.

TT#01

Course: Software Architecture and Design Patterns (SWE 321)

Marks: 20

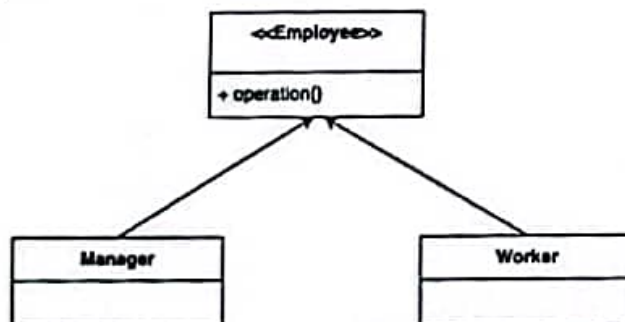
Time: 35 mins

1. What are the potential consequences of violating the **Liskov Substitution Principle** in a software system? How can such violations lead to unexpected behavior or errors? - 04

Builder 2. In a vacation package management system, the development team has employed a design approach that promotes flexibility and modularity in creating customized vacation packages. The system allows users to select destinations, accommodations, activities, and additional services to create their ideal vacation package.

Design the above mentioned system with appropriate design pattern. Draw the class diagram and write the code to support your design. - 06

abstract factory 3. We are supposed to create a computer game that has several scenarios. African scenario, Asian scenario, American Scenario. Each scenario has the following components: terrain, 5 trees of the same kind and 2 animals of the same kind. For each scenario, we have different kinds of terrain, trees and animals. To draw the scenario, we just need to draw the components of a scenario. Design the above mentioned system with appropriate design pattern. Draw the class diagram and write the code to support your design. - 06



factory

Suggest a design pattern to build the above human resource management system. Rewrite the UML class diagram with the completed version of it. - 04

TT#02
Marks: 20

Course: Software Architecture and Design Patterns (SWE 321)
Time: 40 mins

1. "Comments represent a failure to express an idea" - describe with an appropriate example. - 03
2. Let us help a company that is looking to build a robot for cooking. The company wants a simple robot that can simply walk and cook. A user can operate a robot using a set of commands via remote control. Currently, a robot can do three things, it can walk, cook, or can be switched off. The company has set protocols to define the functionality of the robot. If a robot is in an "on" state, you can command it to walk. If asked to cook, the state would change to "cook" or if set to "off", it will be switched off. Similarly, when in the "cook" state it can walk or cook, but cannot be switched off. And finally, when in the "off" state it will automatically get on and walk when the user commands it to walk but cannot cook in the off state. Design the above-mentioned system with appropriate design pattern. Draw the class diagram and write the code to support your design. - 10
3. Can you use the use case of the **Memento** design pattern along with the **Command** pattern? If yes, explain with an example. - 04
4. What is the intent behind the **Observer** design pattern? What are the participants of it? - 03

state pattern