

ફેકલ્ટી ઓફ ઇન્ફોર્મેશન, કમ્યુનિકેશન એન્ડ ટેકનોલોજી

ડિપાર્ટમેન્ટ ઓફ કમ્પ્યુટર સાયન્સ

અભ્યાસક્રમ : માસ્ટર ઓફ કમ્પ્યુટર એપ્લિકેશન (M.C.A.) : સત્ર - ૨

વિષય: MCA-204:સોફ્ટવેર ડીઝાઇન પેટર્ન (GOF)

વસંત પંચમી સત્રાંત પરીક્ષા : મે-2025,

તા. : 03/05/2025

સમય : 08-00 થી 10-30

વાર : શનિવાર

કુલગુણ : 60

લર્નિંગ આઉટકમ

- CO1. Gain a comprehensive knowledge of object-oriented analysis, design, modelling and understanding of design patterns, specifically the GoF design patterns.
- CO2. Learn to recognize common software design problems and apply appropriate design patterns to solve them effectively.
- CO3. Enhance the ability to design software systems that are flexible, scalable, maintainable, and reusable through the application of design patterns.
- CO4. Practice applying design patterns in real-world scenarios and case studies using Java or any other OOP to understand their practical use and benefits.
- CO5. Understand the principles behind design patterns, such as SOLID principles (Single Responsibility, Open-Closed, Liskov Substitution, Interface Segregation, Dependency Inversion), and how they relate to design pattern usage.
- CO6. Improve software quality by designing systems that are more modular, easier to maintain, and less prone to errors through the application of proven design patterns.
- CO7. Develop the ability to critically analyse software designs and evaluate when to apply specific design patterns based on trade-offs, constraints, and requirements.
- CO8. Enhance skills in communicating design decisions using design patterns effectively and documenting patterns used in the software architecture.

પ્રશ્ન-1 (અ) નીચના પ્રશ્નોના જવાબ આપો. (આઠ માંથી ગમે તે છ)

(06)

1. Which is the first step in the Object Oriented Design Process?

ઓબ્જેક્ટ ઓરિએન્ટેડ ડિઝાઇન પ્રક્રિયાનું પહેલું પગથિયું કયું છે?

A) Identify Classes B) Define Relationships C) Decompose classes D) Define System Architecture

2. _____ Keeps code modular and reusable by hiding internal details and exposing only what is necessary.

_____ આંતરિક વિગતોને છુપાવીને ફક્ત જરૂરી માહિતી જ બહાર બતાવવાથી કોડને મોડ્યુલર અને પુનઃવાપરાયોગ્ય બનાવે છે.

3. _____ Use a single interface to represent different types of objects, allowing flexible and reusable code.

_____ અલગ પ્રકારના ઓબ્જેક્ટ્સ માટે એક જ ઇન્ટરફેસ ઉપયોગ કરી શકાય છે, જે કોડના પુનઃઉપયોગ માટે સરળતા આપે છે.

4. _____ Focuses on what an object does instead of how it does it, enabling developers to build on top of existing logic without rewriting it.

_____ કોઈ ઓબ્જેક્ટ શું કરે છે એ પર ધ્યાન આપે છે, કેવી રીતે કરે છે એ નહીં - જેના કારણે હાલના લોજિકના ઉપર નવું ડેવલપ કરી શકાય છે.

5. What is a design pattern in software engineering?

સોફ્ટવેર ઇન્જિનિયરિંગમાં ડિઝાઇન પેટર્ન શું છે?

- A. A specific programming language B. A reusable solution to a common problem
C. A hardware design method D. A new programming paradigm

6. Why are design patterns important in software development?

સોફ્ટવેર ડેવલપમેન્ટમાં ડિઝાઇન પેટર્ન શા માટે મહત્વપૂર્ણ છે?

- A. They increase code duplication B. They reduce performance C. They provide proven solutions and improve code maintainability D. They are used only in hardware design

7. What do design patterns primarily describe in software development?

સોફ્ટવેર ડેવલપમેન્ટમાં ડિઝાઇન પેટર્ન મુખ્યત્વે શું વર્ણવે છે?

- A. Programming languages B. Hardware architecture
C. Software structure and interaction of classes D. Testing tools

8. Which of the following is a benefit of using design patterns?

ડિઝાઇન પેટર્નનો ઉપયોગ કરવાથી નીચેનામાંથી કયો લાભ થાય છે?

- A. Increased code redundancy B. Slower software development C. More flexible and reusable software D. Reduced collaboration

(બ) નીચના પ્રશ્નોના ત્રણ થી ચાર વાક્ય માં જવાબ આપો. (ચાર માંથી ગમે તે બે)

(04)

1. Explain how design patterns help in collaboration during software development.

સોફ્ટવેર ડેવલપમેન્ટ દરમિયાન સહકાર માટે ડિઝાઇન પેટર્ન કેવી રીતે મદદરૂપ થાય છે તે સમજાવો.

2. What are the consequences of using design patterns in software?

સોફ્ટવેરમાં ડિઝાઇન પેટર્નનો ઉપયોગ કરવાથી શું પરિણામો થાય છે?

3. Why is a design pattern called a solution in context?

ડિઝાઇન પેટર્નને "પરિસ્થિતિમાં ઉકેલ" કેમ કહેવાય છે?

4. Write any two Limitations of OOP. OOPની કોઈપણ બે મર્યાદાઓ જણાવો.

(ક) વિગતવાર સમજાવો. (બે માંથી ગમે તે એક)

(05)

1. Find out the problem context and design pattern-based solution and its class diagram from the following statement.

નીચેનાં વિધાનમાંથી સમસ્યાની પરિસ્થિતિ, ડિઝાઇન પેટર્ન આધારિત ઉકેલ અને તેનું ક્લાસ આલેખ કરો

You are building a text editor application where you want to provide users with the ability to undo and redo their actions, such as typing or deleting text. You have to ensure the undo and redo functionality works correctly and efficiently by storing and managing the history of these user actions.

તમે એક ટેક્સ્ટ એડિટર એપ્લિકેશન બનાવી રહ્યા છો, જેમાં યુઝરને ટાઇપિંગ કે ટેક્સ્ટ ડિલીટ કરવા જેવી ક્રિયાઓ માટે undo અને redo કરવાની ક્ષમતા આપવી છે. આ માટે, તમારે યુઝરની આ ક્રિયાઓના ઇતિહાસને સાચવીને અને સંચાલિત કરીને undo અને redo ફંક્શન યોગ્ય અને અસરકારક રીતે કાર્ય કરે તે સુનિશ્ચિત કરવું પડશે.

2. Explain the each Term of GOF design Pattern Template.

ડિઝાઇન પેટર્નના ટેમ્પલેટની દરેક ટર્મ લખો અને સમજાવો.

પ્રશ્ન-2 (અ) નીચના પ્રશ્નોના જવાબ આપો. (આઠ માંથી ગમે તે છ)

(06)

1. _____ shows the classes and objects participating in the pattern.
પેટર્નમાં જોડાયેલા ક્લાસ અને ઓબ્જેક્ટ્સને _____ માં દર્શાવવામાં આવે છે.
2. The _____ of a pattern describes its goal and purpose.
પેટર્નનો _____ એ લક્ષ્ય અને ઉદ્દેશ બતાવે છે.
3. Which section describes the interaction between classes in the pattern?
પેટર્નમાં ક્લાસ વચ્ચેના પરસ્પર સંબંધો કયા વિભાગમાં દર્શાવાય છે?
4. Which section provides real-life examples where the pattern is used?
કયો વિભાગ પેટર્નના વાસ્તવિક ઉદાહરણો આપે છે?
5. _____ is the ability of an object to take many forms.
_____ એ કોઈ ઓબ્જેક્ટની ઘણી જુદી જુદી આકાર ધારણ કરવાની ક્ષમતા છે.
6. _____ allows bundling data and methods that operate on the data within one unit
_____ માહિતી અને તેના પર કામ કરતા પદ્ધતિઓને એકમ તરીકે સંકળવાની રીત છે.
7. _____ helps to hide the complex implementation details and show only essential features to the user.
_____ જટિલ અમલ વિગતોને છુપાવીને વપરાશકર્તાને ફક્ત આવશ્યક લક્ષણો દર્શાવવામાં મદદ કરે
8. In OOP, abstraction is achieved using _____ and abstract classes.
OOP માં, abstraction ની પ્રાપ્તિ _____ અને abstract classes ના ઉપયોગ દ્વારા થાય છે.

(બ) નીચના પ્રશ્નોના ત્રણ થી ચાર વાક્ય માં જવાબ આપો. (ચાર માંથી ગમે તે બે)

(04)

1. What is included in the "Participants" section of a design pattern?
ડિઝાઇન પેટર્નના "Participants" વિભાગમાં શું સમાવવામાં આવે છે ?
2. Write the differences between the Abstract class and the Interface.
Abstract class અને Interface વચ્ચે નો તફાવત લખો.
3. Discuss the importance of Design Patterns from a Java programmer's perspective.
Java પ્રોગ્રામરનાં દૃષ્ટિકોણથી ડિઝાઇન પેટર્નના મહત્વની ચર્ચા કરો
4. Explain, Why GOF design Pattern is divided into three categories?
GOF ડિઝાઇન પેટર્નને ત્રણ શ્રેણીઓમાં શા માટે વિભાજિત કરવામાં આવ્યા છે? સમજાવો.

(ક) વિગતવાર સમજવો. (બે માંથી ગમે તે એક)

(05)

1. Explain the Three Different categories of GOF design patterns.
GOF ડિઝાઇન પેટર્ન્સના ત્રણ વિવિધ શ્રેણીઓનું સમજૂતીપૂર્વક વર્ણન કરો.
2. Explain design pattern role in software development.
સોફ્ટવેર ડેવલપમેન્ટમાં ડિઝાઇન પેટર્ન ભૂમિકા સમજાવો.

પ્રશ્ન-3 (અ) નીચના પ્રશ્નોના જવાબ આપો. (આઠ માંથી ગમે તે છ)

(06)

1. _____ Pattern ensures that only one instance of a class is created.
_____ પેટર્ન ખાતરી આપે છે કે ક્લાસનો માત્ર એક જ ઇન્સ્ટન્સ બનાવાય.
2. The _____ Pattern provides an interface for creating families of related objects.
_____ પેટર્ન સંબંધિત ઓબ્જેક્ટના સમૂહ બનાવવા માટે ઇન્ટરફેસ આપે છે.
3. _____ Pattern allows a class to delegate instantiation to its subclasses.
_____ પેટર્ન ક્લાસને ઇન્સ્ટન્સ બનાવવાની જવાબદારી તેના સબક્લાસને સોંપે છે.
4. _____ Pattern is used to create a copy of an existing object.
પહેલાથી હાજર ઓબ્જેક્ટની નકલ બનાવવા માટે _____ પેટર્ન વપરાય છે.

5. The _____ Pattern separates the construction of a complex object from its representation.
મુશ્કેલ ઓબ્જેક્ટના નિર્માણને તેની રજૂઆતથી અલગ કરવા માટે _____ પેટર્ન વપરાય છે.
6. Creational patterns deal with the process of _____ objects.
ક્રિએશનલ પેટર્ન ઓબ્જેક્ટ બનાવવાની પ્રક્રિયા સાથે સંકળાયેલા હોય છે.
7. Creational Design Patterns are used to _____ the process of object creation, making the system more flexible and reusable.
ક્રિએશનલ ડિઝાઇન પેટર્ન્સનો ઉપયોગ ઓબ્જેક્ટની બનાવટની પ્રક્રિયાને _____ કરવા માટે થાય છે, જેથી સિસ્ટમ વધુ લવચીક અને પુનઃવાપરવા યોગ્ય બને.
8. _____ Pattern is commonly used when object creation is costly.
જ્યારે ઓબ્જેક્ટ બનાવવાની પ્રક્રિયા મોંઘી હોય ત્યારે સામાન્ય રીતે _____ પેટર્ન વપરાય છે.

(બ) નીચના પ્રશ્નોના ત્રણ થી ચાર વાક્ય માં જવાબ આપો. (ચાર માંથી ગમે તે બે)

(04)

1. When is the Prototype pattern preferred? / પ્રોટોટાઇપ પેટર્ન ક્યારે ઉપયોગી બને છે?
2. What is the difference between Factory Method and Abstract Factory?
ફેક્ટરી મેથડ અને એબ્સ્ટ્રેક્ટ ફેક્ટરી વચ્ચે શું તફાવત છે?
3. Why should cloning be prevented in a Singleton class? Singleton?
ક્લાસમાં ક્લોનિંગને રોકવું શા માટે જરૂરી છે?
4. Why is thread safety important in Singleton design pattern? Singleton
ડિઝાઇન પેટર્નમાં થ્રેડ સેફ્ટી શા માટે મહત્વપૂર્ણ છે?

(ક) વિગતવાર સમજવો. (બે માંથી ગમે તે એક)

(05)

1. Explain why the builder and factory design patterns are related.
Builder અને Factory ડિઝાઇન પેટર્ન્સ શા માટે એકબીજાથી સંબંધિત છે? સમજાવો
2. Find out the right design pattern-based solution and its class diagram from the following statement.
નીચેનાં વિધાનમાંથી સમસ્યા માટે સાચી ડિઝાઇન પેટર્ન આધારિત ઉકેલ અને તેનું ક્લાસ આલેખ કરો

Problem:

Imagine you are designing a system for a game that has different types of characters. Each character has unique attributes and behavior. You want to allow the game to create different types of characters (like Warrior, Mage, Archer) based on user input, but you don't want the user to be involved in the details of how these characters are created.

કલ્પના કરો કે તમે એક ગેમ માટે સિસ્ટમ ડિઝાઇન કરી રહ્યા છો જેમાં વિવિધ પ્રકારના પાત્રો છે. દરેક પાત્રના વિશિષ્ટ ગુણ અને વર્તન છે. તમે ઇચ્છતા છો કે ગેમ યુઝર ઇનપુટના આધારે વિવિધ પ્રકારના પાત્રો (જેમ કે વોરિયર(યોધા), મેજ (જાદુગર), આર્ચર(તીરંદાજ)) બનાવી શકે, પરંતુ તમે નથી યાહતા કે યુઝરને પાત્રો બનાવવાના વિગતોમાં સામેલ કરવો.

પ્રશ્ન-4 (અ) નીચના પ્રશ્નોના જવાબ આપો. (આઠ માંથી ગમે તે છ)

(06)

1. The _____ pattern allows the client to interact with complex subsystems through a simplified interface.
_____ પેટર્ન ક્લાયંટને સરળ ઇન્ટરફેસ દ્વારા જટિલ સબસિસ્ટમ સાથે સંપર્ક કરવાનો મોકો આપે છે.
2. The _____ pattern allows different classes to have a common interface, making it easier to work with them interchangeably. _____ પેટર્ન વિવિધ ક્લાસોને એક સામાન્ય ઇન્ટરફેસ પ્રદાન કરે છે, જેથી તેમને પરસ્પર બદલવામાં સરળતા રહે છે.
3. In the _____ pattern, used to add additional responsibilities to an object dynamically. _____ પેટર્નનો ઉપયોગ ઓબ્જેક્ટમાં નવી જવાબદારીઓ ડાયનામિક રીતે ઉમેરવા માટે કરવામાં આવે છે.

4. The _____ pattern allows you to provide a uniform interface for different objects without changing their implementation.
_____ પેટર્ન દ્વારા તમે વિવિધ ઓબ્જેક્ટ્સ માટે એકરૂપ ઈન્ટરફેસ પ્રદાન કરી શકો છો, તેનાં અમલને બદલ્યા વગર.
5. The _____ pattern is used when an object changes its behavior based on its internal state.
_____ પેટર્નનો ઉપયોગ તે સમયે થાય છે જ્યારે એક ઓબ્જેક્ટ તેના આંતરિક અવસ્થાના આધાર પર તેનો વર્તન બદલતો હોય છે.
6. In the _____ pattern, an object keeps a reference to its successors and is responsible for handling a request by passing it along the chain.
_____ પેટર્નમાં, એક ઓબ્જેક્ટ તેના અનુગામી ઓબ્જેક્ટનો સંદર્ભ રાખે છે અને ચેઇનમાં વિનંતીને આગળ મોકલીને તેનો જવાબદારી લે છે.
7. The _____ pattern defines a set of algorithms, encapsulates each one, and makes them interchangeable.
_____ પેટર્ન એક ક્રમિત અલ્ગોરિધમનો સેટ પરિભાષિત કરે છે, દરેકને કેપ્સ્યુલેટ કરે છે અને તેને પરસ્પર વિલય માટે બનાવે છે.
8. The _____ pattern is used when an object (subject) changes state, and all its dependent observers are notified and updated automatically.
_____ પેટર્નનો ઉપયોગ ત્યારે થાય છે જ્યારે એક ઓબ્જેક્ટ (વિષય)ની અવસ્થા બદલાય છે, અને તેના તમામ નિર્ભર નિરીક્ષકોએ આપોઆપ સૂચિત અને અપડેટ થવું છે.

(બ) નીચના પ્રશ્નોના ત્રણ થી ચાર વાક્ય માં જવાબ આપો. (ચાર માંથી ગમે તે બે)

(04)

1. When would you use a Proxy pattern? તમે ક્યારે પ્રોક્સી પેટર્નનો ઉપયોગ કરશો?
2. What is the difference between the Decorator and the subclassing approach?
ડેકોરેટર અને સબક્લાસિંગ પદ્ધતિમાં શું તફાવત છે?
3. What kind of relationship does the Chain of Responsibility pattern establish between the request and the handler? ચેઇન ઓફ રિસ્પોન્સિબિલિટી પેટર્ન વિનંતી અને હેન્ડલર વચ્ચે કયું પ્રકારનો સંબંધ સ્થાપિત કરે છે?
4. How is the State pattern different from the Strategy pattern? Explain
સ્ટેટ પેટર્ન સ્ટ્રેટજી પેટર્નથી કેવી રીતે જુદી છે? સમજાવો

(ક) વિગતવાર સમજાવો. (બે માંથી ગમે તે એક)

05

1. Find out right Design pattern and write consequences, also known as and related pattern for the same. આપેલા સ્ટેટમેન્ટ માટે યોગ્ય ડિઝાઇન પેટર્ન શોધો અને તેના પરિણામો, તેને અન્ય નામે ઓળખવામાં આવે છે તે, તથા સંબંધિત પેટર્ન લખો.
1. In a customer support system, a basic support agent handles simple queries, a supervisor handles more complex issues, and a manager handles critical complaints. The request should move through this chain until it is handled.
એક ગ્રાહક સહાય સિસ્ટમમાં સામાન્ય પ્રશ્નો માટે હેલ્પડેસ્ક, જટિલ પ્રશ્નો માટે સુપરવાઇઝર અને ગંભીર ફરિયાદ માટે મેનેજર જવાબદાર હોય છે. વિનંતી ત્યારે સુધી આગળ વધવી જોઈએ જ્યાં સુધી યોગ્ય વ્યક્તિ તેને સંભાળી ન લે.

OR (અથવા)

1. Find out the right design pattern-based solution and its class diagram from the following statement
નીચેનાં વિધાનમાંથી સમસ્યા માટે સાચી ડિઝાઇન પેટર્ન આધારિત ઉકેલ અને તેનું ક્લાસ આલેખ કરો

Consider an application that handles the creation and management of graphical user interface (GUI) components such as buttons, text fields, and panels. Each component can have additional features, such as borders, scrollbars, and tooltips, that can be added dynamically at runtime.

એક એપ્લિકેશન પરિસ્થિતિને માન્ય રાખો જે ગ્રાફિકલ યુઝર ઇન્ટરફેસ (GUI) ઘટકો જેમ કે બટન, ટેક્સ્ટ ફિલ્ડ અને પેનલ્સને બનાવવામાં અને મેનેજ કરવામાં મદદ કરે છે. દરેક ઘટકમાં સિસ્ટમ રણટાઈમ દરમિયાન બોર્ડર્સ, સ્ક્રોલબાર્સ અને ટૂલટિપ્સ જેવા વધારાની વિશેષતાઓ ઉમેરવામાં આવે છે.