GUJARAT TECHNOLOGICAL UNIVERSITY

Syllabus for Integrated MCA 9th Semester
Branch: MCA Integrated
Subject Name: Design Pattern
Subject Code: 2698601

With effective from academic year 2022-23

Teaching Scheme			Credits	Examination Marks				
				Theory		Tutorial/ Practical		Total Marks
L	P	OJT	C	University exams (ESE)	Progressive Assessment (PA)	External Practical /viva Exam(ESE)	Internal evaluation Practical /viva Exam(PA)	
4	2		6	70	30	30	20	150

Pre-requisites:
1. Object Oriented Analysis and Design Concepts.
2. Data structures and algorithms.
3. Knowledge of Object oriented Programming

Learning Objectives:

1. Recognize the importance of design.
2. Determine object granularity and Specify object interfaces.
3. Summarizes the importance of design patterns.
4. Identify the need of template method and visitor Discussion of Behavioural Patterns.
5. Review the template of design patterns.

Course Outcome (COs):
CO1: Construct a design consisting of a collection of modules.
CO2: Exploit well-known design patterns (such as Iterator, Observer,Factory and Visitor).
CO3: Distinguish between different categories of design patterns.
CO4: Apply common design patterns to incremental/iterative development.
CO5: Identify appropriate patterns for design of given problem.
CO6: Design the software using Pattern Oriented Architectures.

Course Content:

Unit No.	Content	Hrs
1	What is a Design Pattern?, Design Patterns in Smalltalk MVC, Describing Design Patterns, The Catalogue of Design Patterns, Organizing The Cato log, How Design Patterns solve Design Problems, How to Select a Design pattern, How to Use a Design Pattern.	**7**
2	A Case Study: Designing a Document Editor, Design Problems , Document Structure, Formatting , Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations Spelling Checking and Hyphenation, Summary, Creational Patterns, Abstract Factory, Builder , Factory Method, Prototype, Singleton, Discussion of Creational Patterns.	**12**
3	Structural Pattern Part-I, Adapter, Bridge, Composite. Structural Pattern Part-II, Decorator, Facade, Flyweight, Proxy.	**8**

4	Behavioral Patterns Part: I, Chain of Responsibility, Command, Interpreter, Iterator. Behavioral Patterns Part: II, Mediator, Memento, Observer, Discussion of Behavioral Patterns.	**8**
5	Behavioral Patterns Part: III, State, Strategy, Template Method, Visitor, Discussion of Behavioral Patterns. What to Expect from Design Patterns, A Brief History, The Pattern Community, An Invitation, A Parting Thought.	**10**
	Total Hours:	**45**

Suggested Specification table with Marks (Theory):

Distribution of Theory Marks				
R Level	U Level	A Level	N Level	E Level
10	40	35	15	0

Legends: R: Remembrance; U: Understanding; A: Application, N: Analyze and E: Evaluate and above Levels (Bloom's Taxonomy)

Reference Books:

1. Design Patterns By Erich Gamma, Pearson Education
2. Design Patterns Explained By Alan Shalloway, Pearson Education..
3. Meta Patterns designed by Wolf gang, Pearson.
4. Head First Design Patterns By Eric Freeman-Oreilly-spd.
5. Patterns in JAVA Vol-I By Mark Grand, Wiley Dream Tech.
6. Patterns in JAVA Vol-II By Mark Grand, Wiley Dream Tech.

Suggested List of Practical as follows:

1. **Write a java program to demonstrate Singleton Design Pattern considering following scenario:**

Define a class with a method that creates a new instance of the class if one does not exist. The class has instance attribute that is defined private and static Make the constructor of the **class** private so that there is no other way to instantiate the class. Create the accessor method as public and static. The accessor method is for obtaining the reference to the singleton object.

2. **Write a java program to demonstrate Factory design pattern considering following scenario:**

The Restaurant makes various types of traditional Thali for the customer. Create an abstract class thali. In this abstract class we define four abstract methods such as addSabji(), addDal(), addRice(), addRoti(). Also there is a method makeThali() method which prints "Veg Thali will be ready in 30 minutes".Create a class GujaratiThali which extends an abstract class Thali. In this class implement the abstract methods addSabji(), addDal(), addRice(), addRoti(). Create a class PunjabiThali which extends an abstract class Thali. In this class implement the abstract methods addSabji(), addDal(), addRice(), addRoti(). Create an abstract class BaseThaliRestaurant in which implemented an abstract method createThali() having an argument of string type. Create a class ThaliRestaurant which extends BaseThaliRestaurant class. In this class implement an abstract method createThali() which is defined in BaseThaliRestaurant class. In createThali() method implement a switch case for thali such as gujarati thali and punjabi thali. Lastly, create FactoryDesignPattern class from which access methods defined in previous classes and perform certain operations.

3 Write a java program to demonstrate Prototype design pattern considering following scenario:

Create two classes .One is called User that is used for creating the list of users. The second class is a service that is for fetching a list of existing users. Currently it just mocks the service call and loads the data into list by creating it. Create a class UserDetail that implements the Java Clonable interface. This is an interface that enforces the class to override the method clone() which is used to obtain cloned objects. This method uses the existing list of users and creates a new list out of it. It does not go for a service call. Each cloned object instance has the data from the actual object along with the new data that is being added to it.

4 Write a java program to demonstrate Builder design pattern considering following scenario:

There is a food ordering at restaurant. In food ordering customer is acting as a client class, cashier acting as a director class and restaurant crew are acting as a builder class. In the restaurant there are two types of meals : veg meal and non veg meal. Veg meal consists of veg burger, fries, coke and toy car whereas Non veg meal consists of non veg pizza, fries, Pepsi and toy bike. Construction process for both the meal are same and it consists of five steps such as main item, side item, drink, toy and then pack. If suppose a customer went to the restaurant and orders a veg kid meal to the cashier. Cashier then forwards the order to the restaurant crew to complete the order. Now, restaurant crew first build the veg burger then they will build fries then coke and at last they build toy. After building all the meal restaurant crew now will pack the meal and forwards to the customer.

5 Write a java program to demonstrate bridge design pattern considering following scenario:

Create two implementations of a bridge class. These implementers assist in doing the tasks that the bridge class was created for. Thus, they utilise the abstract class to further advance the code and provides specific functionalities to the caller class. Although the code is self-explanatory, we would try and understand the same using the below images. Let us look at the below images. These images explain how the implementation would have been in absence of bridge pattern and how it changed by implementing using the bridge design pattern.

6 Write a java program to demonstrate composite design pattern considering following scenario:

Create a property interface that has two basic functionalities – Buy and sell. Every property item can be bought as well as sold. Create three different classes – Apartments, Bungalow and Tenaments. These three different classes can implement the property class created as a component interface. The above three classes are called as leaf classes as they implement a component interface. Finally, create a class called Application that uses a collection of different types of property.

(Note: there can be four files of this program: property.java , Apartments.java, Bungalow.java, Tenaments.java and Application.java)

7 Write a java program to demonstrate Adapter design pattern considering following scenario:

Create Consider a class Flour with two attributes – weight and price. Create a class FlourItem with a method to get the object of Flour class with default values. Create an Interface FlourItemInterface that contains 3 different methods to provide flour packets of 3 different sizes (for example: public Flour getQuintal(); public Flour get10kg();public Flour get1kg();). Implement the interface and set the relevant pricing.

8 Write a java program to demonstrate Proxy design pattern considering following scenario:

Create an interface called SchoolInternet in which define an abstract method called provideInternet().Create a class RealInternet that implements SchoolInternet interface. In this class create one private object departmentName and also create a method RealInternet() .Implement a method provideInternet() in this class. Create a class called ProxyInternet which implements SchoolInternet interface. In this class create one private object departmentName and one RealInternet class object. Lastly create ProxyDesignPattern class from which access different methods created in previous classes

9 Write a java program to demonstrate Decorator design pattern considering following scenario:

A pizza company make an extra topping calculator. A user can ask to add extra topping to a pizza and chef will add toppings and increase its price using the system.

10 Write a java program to demonstrate observer design pattern considering following scenario:

Create a class Subject that includes basic three functions register, unregister, notifyAllObservers. The interface named Observer has only one abstract method called update().Create Celebrity class and Follower class that implement Observer.Also implement ObserverDesignPattern Class that includes instances (objects) f Celebrity class and Follower class.

11 Write a java program to demonstrate Mediator design pattern considering following scenario:

Road traffic police officer (RTPC) is a mediator between vehicles. It helps in communication between vehicles and co-oridates/controls and allows vehicles to stop or go forward. Two vehicles does not interact directly and there is no dependency between them. This dependency is solved by the mediator (Road traffic police officer).

Create a class RTPCMediator that implements IRTPCMediator interface. In the class, create several methods such as registerRoad(), registerVehicle(), isStarted(), isStopped(), setStartedStatus(), setStoppedStatus(). You can also add several abstract classes if required. Write a code to demonstrate above mentioned case.

12 Write a java program to demonstrate Chain of responsibility Design pattern considering following scenario:

Torrent Bill Payment Kiosk where customers can pay their electricity bill instead of standing in the queue. If any customer wants to pay his/her electricity bill via kiosk then he/she has to enter their customer number and have to place their money in cash collector. Suppose the amount of money is 1745 then inside the Kiosk 500 rupees handler performs operation and fetches two three notes of five hundred and passes the request to 100 rupees handler it performs operation and fetches two notes of one hundred. Similarly for 10 rupees handler it fetches four notes of ten rupees and 5 rupees handler it fetches one note of five rupees.

13 Write a java program to demonstrate Iterator design pattern considering following scenario:

Create a class Shape .It has attributes id and name .Create a class ShapeStorage store the Shape objects. This class contains an array of Shape type. The addShape method is used to add a Shape object to the array and increment the index by one. The getShapes method returns the array of Shape type. Create class ShapeIterator that is an Iterator to the Shape class. This class implements the Iterator interface and defines all the methods of the Iterator interface. The has Next method returns a boolean if there's an item left.

The next method returns the next item from the collection and the remove method remove the current item from the collection.

14 Write a java program to demonstrate State design pattern considering following scenario:

There is a company which 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. A robot can do three things: it can walk, cook, or can be switched off.
If a robot is in "on" state it can accept command to walk. If asked to cook, the state will change to "cook" or if set to "off", it will be switched off. Similarly, when in "cook" state it can walk or cook, but cannot be switched off. And finally, when in "off" state it will automatically get on and walk when the user commands it to walk but cannot cook in off state.

15 Write a java program to demonstrate Template design pattern considering following scenario:

We want to make a bru coffee for one of our guests. The making of bru coffee includes boiling water, adding milk, adding sugar as required and then finally adding bru coffee to it. Some another guest demanded for nescafe coffee instead of bru coffee. Since, the process of preparing coffee is same and just the adding of coffee powder varies as per guest demand, create a template method called prepareCoffee() and define the steps for making the coffee either bru coffee or nescafe coffee By using this template method we can make any flavour or any type of coffee. Hence, we can consider this example is a best real life example to understand Template design pattern.