Course: 302: **Design Patterns**

<table>
<tr><td>Course Code</td><td>302</td></tr>
<tr><td>Course Title</td><td>Design Patterns</td></tr>
<tr><td>Credit</td><td>4</td></tr>
<tr><td>Teaching per Week</td><td>4 Hrs.</td></tr>
<tr><td>Minimum weeks per Semester</td><td>15 (Including Class work, examination, preparation, holidays etc.)</td></tr>
<tr><td>Review / Revision</td><td>June 2021</td></tr>
<tr><td>Purpose of Course</td><td>The purpose of the course is to make student understand how Patterns can be implemented in various object oriented programming languages to solve real world problems.</td></tr>
<tr><td>Course Objective</td><td>The objective of the course is -
1. To study various Design Patterns
2. How these Patterns can be used to design better systems through Object Oriented Programming Languages</td></tr>
<tr><td>Course Outcome</td><td>CO1: Explain students about the various design patterns; their categories, and purpose.
CO2: Explain the creational design patterns.
CO3: Explain the structural design patterns.
CO4: Explain the behavioural design patterns.
CO5: Explain some more design patterns used in IT industry currently.
CO6: Make students understand the applicability of design patterns practiced by IT companies and how effectively combine these patterns for effective software development.</td></tr>
<tr><td>Mapping between COs with PSOs</td><td>
<table>
<tr><th></th><th>PSO1</th><th>PSO2</th><th>PSO3</th><th>PSO4</th><th>PSO5</th><th>PSO6</th><th>PSO7</th><th>PSO8</th></tr>
<tr><td>CO1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO3</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>CO6</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Pre-requisite</td><td>Object Oriented Programming, Software Engineering</td></tr>
<tr><td>Course Content</td><td>Unit -1 Creational Patterns
1.1 Singleton Pattern
1.2 Prototype Pattern
1.3 Builder Pattern
1.4 Factory Method Pattern
1.5 Abstract Factory Pattern

Unit-2 Structural Patterns
2.1 Proxy Pattern
2.2 Decorator Pattern
2.3 Adapter Pattern
2.4 Façade Pattern
2.5 Flyweight Pattern
2.6 Composite Pattern
2.7 Bridge Pattern

Unit-3 Behavioural Pattern
3.1 Visitor Pattern
3.2 Observer Pattern
3.3 Strategy Pattern
3.4 Template Method Pattern
3.5 Command Pattern
3.6 Iterator Pattern
3.7 Memento Pattern
3.8 State Pattern</td></tr>
</table>

	3.9 Mediator Pattern 3.10 Interpreter Pattern **Unit-4 Additional Design Patterns** 4.1 Simple Factory Pattern 4.2 Null Object Pattern 4.3 MVC Pattern **Unit-5 Pattern Applicability** 5.1 Security Patterns Repository 5.2 Patterns for Agile Development 5.3 Restful Service Patterns 5.4 Solution with semaphore 5.5 Patterns and Pattern combination in practice 5.6 Big Ball of Mud **Self-Study :** Pattern Languages
Reference Books	1. Design Patterns: Elements of Reusable Object-Oriented Software, Erich Gamma, Richard Helm, Ralph, John, Addision Wesley 2. Head First Design Patterns, Eric Freeman, O'Reilly 3. Design Patterns in C#, Vaskaran Sarcar, Apress 4. Design Patterns in Modern C++, Reusable Approaches for Object-Oriented Software Design, Dmitri Nesteruk, Apress 5. Modern C++ design: generic Programming and design patterns applied, Alexendrescu, Andrei, Addison-Wesley 6. Java Design Patterns: A Hands-on Experience with Real-World Examples, Vaskaran Sarcar, Apress
Teaching Methodology	Class work, Discussion, Self-Study, Seminars and/or Assignment
Evaluation Method	30 % internal assessment and 70% external assessment