|  |  |
| --- | --- |
| **Course Outline:** |  |
|  | Introduction, MVC, Roles, Interface-vs-Implementation |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Clone Idiom |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Prototype Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Factory Method Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | How to disable copying and assignment in a class |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Singleton Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | Comparing Prototype, Factory Method, and Singleton |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Abstract Factory Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Bridge Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Proxy Design Pattern, Auto Pointer, Reference Counting |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Iterator Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Composite Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Flyweight Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Interpreter Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Visitor Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Chain of Responsibility Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Memento Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Command Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The All Of Me Idiom |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The State Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Decorator Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. Solution - a description of the "best practice" way to solve that problem. Setup - C++ code sample that creates objects. Use - C++ code sample that uses objects. Classes - C++ header files and source files for the classes in the solution. Design Pattern - a description of what makes this a "design pattern" When To Use (this pattern) Makes It Easier To ... Reuses (previous design patterns) References to the literature Lab instructions |  |
|  | The Observer Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Adapter Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Facade Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Template Method Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Mediator Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |
|  | The Strategy Design Pattern |  |
|  | * Problem - a description of a common software engineering problem. * Solution - a description of the "best practice" way to solve that problem. * Setup - C++ code sample that creates objects. * Use - C++ code sample that uses objects. * Classes - C++ header files and source files for the classes in the solution. * Design Pattern - a description of what makes this a "design pattern" * When To Use (this pattern) * Makes It Easier To ... * Reuses (previous design patterns) * References to the literature * Lab instructions |  |