

## **ECE495K OBJECT-ORIENTED SCRIPTING AND OBJECT-ORIENTED DESIGN**

**Sem 2, Class 3, Credit 3**

**Pre-requisite:** ECE462

### **Prerequisite By Topic:**

A student wishing to take this course must already be proficient with object-oriented programming. The student must possess a deep-level understanding of the roles played by polymorphism, inheritance, and encapsulation in object-oriented programs. A student must also possess a strong background in working with pointers and class hierarchies.

### **Engineering Design Content:**

Designing object-oriented programs of specified functionality with systems programming languages and scripting languages. Constructing test cases for the evaluation of such programs.

### **Engineering Design Considerations:**

Development of extendible and maintainable software using object-oriented concepts.

### **Justification:**

This course takes a student beyond the syntax of OO programming and into the realm of design of OO software. In its broadest sense, OO design deals with issues ranging from how to give object-orientation to a new problem domain to the use of the design patterns that represent template solutions to the most frequently encountered problems in writing OO programs. The course will explore the issues of giving OO orientation to new problems and the use of design patterns for both systems programming languages and for scripting languages. C++ and Java will be used for systems programming languages, and Perl and Python will be used for scripting languages.

### **Course Description:**

Unified modeling language. Use case analysis. Constructing conceptual models. System sequence diagrams. "Gang of Four" design patterns. Case studies. A brief overview of C++ and Java for systems programming languages. An overview of Perl and Python for scripting languages. Object-oriented programming with Perl and Python. The notion of an object reference in Perl and Python. Blessing object references in Perl. Inheritance and polymorphism with Perl and Python. Object persistence and other database programming issues for both systems programming languages and scripting languages.

### **Course Outcomes:**

A student who successfully fulfills the course requirements will have demonstrated:

- i) a knowledge of the Unified Modeling Language for the conceptual design of object-oriented programs. (3, 4, e, k)
- ii) an ability to design object-oriented solutions to programming problems using previously developed "best practice" design components. (3, 4, c, e, k)
- iii) object-oriented programming with Perl and Python. (3, 4, e, k)
- iv) designing object-oriented software in Perl and Python for scripting languages and in C++ and Java for systems programming languages. (3, 4, c, e, k)
- v) ability to write object-oriented database programs in C++ and Java. (3, 4, e, k)
- vi) ability to write object-oriented database programs in Perl and Python. (3, 4, e, k)
- vii) ability to write object-oriented socket programs in C++ and Java. (3, 4, e, k)
- viii) ability to write object-oriented socket programs in Perl and Python. (3, 4, e, k)

**Text:**

- 1 "Design Patterns, Elements of Reusable Object-Oriented Software," by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. Addison-Wesley. 1994. ISBN 0-201-63361-2.
- 2 "Advanced Perl Programming," by Sriram Srinivasan. O'Reilly, 1997. ISBN 1-56592-220-4.

**References:**

"Programming with Objects, A Comparative Presentation of Object-Oriented Programming with C++ and Java" by A. C. Kak, John-Wiley, March 2003, ISBN 047-126-8526.

| <b>Lecture</b> | <b>Topic</b>                                                         |
|----------------|----------------------------------------------------------------------|
| 1              | Course Introduction                                                  |
| 2              | Software Development Process for Large OO Programs                   |
| 3              | Use Cases, Class, Interaction, Package, State, and Activity Diagrams |
| 4-8            | Overview of Perl and Python                                          |
|                | Test 1                                                               |
| 9-16           | Object-Oriented Programming with Perl and Python                     |
| 17-25          | Design Patterns                                                      |
| 26-32          | Implementations of Design Patterns in C++ and Java                   |
|                | Test 2                                                               |
| 33-35          | Implementation of Design Patterns in Perl and Python                 |

|       |                             |
|-------|-----------------------------|
| 33-35 | OO for Network Programming  |
| 36-38 | OO for Database Programming |
| 39-42 | Distributed OO Programming  |
|       | Test 3                      |

**Assessment Methods for Course Outcomes:** Each of the outcomes will be assessed by giving the students appropriate C++ and Java programming assignments.