

CS441 - Current Topics in Programming Languages

Last Updated - 04/15/02

Course Manager – Dr. Tzilla Elrad, Research Professor

3 credit hours; elective for CS & CPE; 150 min. lecture each week

Current Catalog Description – New topics in programming language design such as concepts of concurrent and distributed programming, communicating sequential processes, and functional programming. System development tools and language features for programming. An introduction to programming language semantics. Prerequisite: CS 331 or CS 401 or CS 403. (3-0-3)

Textbook

- Deitel & Deitel, *Java- How to Program*, Fourth edition, 2001 Prentice Hall.
- Deitel and Deitel, *Advanced Java how to Program*, First Edition. 2001 Prentice Hall.

References - other textbooks or materials

- none

Course Goals

The course is basically language independent. Any language that can support the course goals may be selected. An example of a choice language might be Java.

- Outline the evolution of the architectural neutral, secure, OO programming languages in order to illustrate how this evolution has led to the occurrence of the JAVA programming model. The course builds on the students' knowledge of Object Oriented Programming concepts, which is a prerequisite for the course.
- Design, implement, test, and debug Applets, Servlets, and Applications.
- Design and implement Graphical User Interfaces.
- Learn the programming language mechanisms that support distribution transparency and development of distributed applications.
- Recognize the underlying concurrency language model; Multithreading and monitor-based concurrency model.
- Demonstrate the supportive language constructs and mechanisms for the design and development of 3-tier architectures; server-side programming.

Prerequisites by Topic

- Strong object-oriented programming experience.

Major Topics Covered in Course

1. The evolution of the architecture neutral languages and Abstract Computing Machines	1 hour
2. Structure of Virtual Machines: Performance vs. Portability	2 hours
3. The Input/Output Framework: byte/character streams and stream chaining	3 hours
4. Control and data abstractions	4 hours
5. Object Oriented Programming Constructs	2.5 hours
6. Garbage Collectors	1 hour
5. Event-driven programming for building GUI	6 hours
6. Exception Handling	2 hours
7. Multithreading and concurrency support	5 hours
8. API Support for SQL: Java Data Base Connectivity	3.5 hours
9. Server-side programming: Servlets and JSP	4.5 hours
10. Distributed computing: Network programming and RMI middleware	4 hours
11. Reflection Support	1 hour
12. Post object orientation approaches - Aspect Oriented Software Development.	6 hours
Total hours	45 hours

Laboratory projects (specify number of weeks on each)

- The students will have 4 programming assignments. Each assignment demonstrates the important concepts covered in lectures and 1 individual project.
 - Homework1: Write a Java program to understand and practice the Virtual Machine programming Model
 - Homework2: Design and implement a Java program to utilize numerous Java applications frameworks.
 - Homework3: Write an OO application that utilizes majority of the language constructs that support object-orientation.
 - Homework4: Write a GUI to practice the event-driven programming model for building GUI
 - Project: Design and develop a 3-tier application that utilizes the design principles and languages constructs that are covered in the course.

Estimate CSAB Category Content in Credit Hours

	CORE	ADVANCED		CORE	ADVANCED
Data Structures			Computer Organization and Architecture		
Algorithms			Concepts of Programming Languages		2
Software Design		1			

Oral and Written Communications - Every student is required to submit at least __0__ written reports (not including exams, tests, quizzes, or commented programs) of typically __0__ pages and to make __0__ oral presentations of typically __0__ minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

- none

Social and Ethical Issues - Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

- none

Theoretical Foundations - Please list the types of theoretical material covered, and estimate the time devoted to such coverage in contact (lecture and lab) hours.

- The concept of abstract computing machines, the virtual machines, is discussed along with the linguistic constructs to support distribution transparency, multithreading and concurrency control.

Problem Analysis - Please describe the problem analysis experiences common to all course sections.

- Understand and classify programs based on their application domains: application, client-side, and server-side programs.
- Recognize concurrency requirements and the mechanisms by which these requirements can be codified.
- Identify distribution transparency requirements and available middleware support.
- Review application requirements and understand the supportive application frameworks to meet these requirements.

Solution Design - Please describe the design experiences common to all course sections.

- Deploy OO design principles in the development of Applications, Applets, and Servlets
- Utilize the application frameworks in the software development process.
- The use of RMI as the middleware vehicle in the design of the distributed applications
- Apply common concurrency patterns to build thread-safe applications.

Other Course Information

- Planned Course Enhancements
 - Update textbook to be less Java reference and more object-oriented theory based. Possible choice Czarnecki, K., Eisenecker, U. W., Generative Programming: Methods, Techniques, and Applications, Addison-Wesley, 1999.
 - Update Catalog Description - Topics in programming language design such as concepts of abstract computing machines and virtual machines, Multithreading and distributed programming, garbage collector, even-driven programming, language framework for design and development of graphical user interface, and the supportive language constructs and mechanisms for the design and development of 3-tier software architectures. Current topics in programming languages such as issues in post object orientation approaches - Aspect Oriented Software Development. Prerequisite: CS 331 or CS 401 or CS 403. (3-0-3) (Fall 2002)
 - The CS Undergraduate Committee is considering changing the introductory programming language taught to CS and CPE majors to Java in the near future and to stress problem solving and object oriented concepts more heavily. This change may effect CS441 as most of the first 5 "Major Topics Covered in Course" would be covered in pre-requisite classes which would present an opportunity for CS441 to contain more advanced theoretical topics.
 - Review of CS441, CS445, CS447 for possible overlap, better transition, and increase in broader theoretical topics.
 - Introduce more Aspect Oriented Software Development - After years of experience with object oriented programming and the advances in theory and applications of OO design and development it possible to identify the limitations of object orientation. "Post Object Orientation" approaches to software development that extend the basic ideas behind OO are quite mature. Aspect Orientation is emerging as an attempt to unify these new approaches. Scientific and industrial experiences are already available. The course will cover the areas where object orientation fails to adequately express separation of concerns and will cover the basic extension provided by aspect orientations. Applications to database, networking, concurrent and distributed programming, real-time application and electronic commerce will be presented.