20-ECES-694 ADVANCED PROGRAMMING TECHNIQUES

Catalog Data: 20-CS-694. Advanced Programming Techniques. Credits 3. Treats programming concepts, especially those needed for upcoming internet, security, graphics, and stream applications. Topics include: exception handlers, control abstractions and applications to streams, co-routines, demand- driven computation, interfaces substitution for multiple inheritance, content handlers, protocol handlers, applications of public key cryptosystem, clients and servers, remote method invocation, uniform data base interfaces, and reflection. Prereq: 20-CS-403, 20-EECE-429.

Textbook: P. Niemeyer and J. Peck, *Exploring Java*, *3*nd Ed., O'Reilly, 2005.
References: P. Deitel and H. Deitel, *Java How to Program*, *6*th Ed., Prentice-Hall 2004.
Coordinator: John Franco, Professor, Computer Science.

Goals: The programming and problem solving skills of a typical student are expected to be sharpened to the point where a student can be called a "crackerjack coder". The advanced use of exceptions, threads, reflection, streams, sockets, remote method invocation, virtual functions, among other concepts will be considered in our study. In addition, modern security features including provisions for public-key cryptosystems and class encoding will be discussed. All the examples we use will be coded in Java. Hence, learning Java will be part of the course.

Prerequisites by Topic:
20-CS-403, Organization of Programming Languages
20-EECE-429, Introduction to Operating Systems
Sophistication in the use of object oriented languages.

Topics:

1. Translations from C++ to Java (2 classes)
2. Exceptions (2 classes).
3. Threads : Streams, co-routines, etc. (8 classes)
4. Input and output streams (1 class)
5. Networking : Sockets, RMI etc. (6 classes)
6. Computer and Network Security (3 classes)
7. Reflection (3 classes)
8. Data Base Engines (2 classes)
9. Graphics and Animation (2 classes)

Computer Usage:

1. Students use a UNIX and/or Windows platform SUN or IBM Java compiler and an editor.
2. All five homeworks plus the term project are code-writing exercises.

Lab Projects: All homeworks and a term project

Outcomes Addressed:
. An ability to design a system, component, or process to meet desired needs.
. An ability to identify, formulate, and solve problems that arise in their discipline and its applications.
. An ability to use the techniques, skills, and modern tools of the discipline necessary for professional practice.

Area Coverage:

AREA	**HOURS**
Algorithms	2
Data Structures	4
Comp. Organization and Architecture	0
Software Design	7
Concepts of Programming Languages	17

Contributions to CS Student Outcomes:

- A final project that is complex and includes security and safety measures as well as java constructs such as threads ensures the student will be able to design a complex system with many components.
- The midterm examination ensures the student can solve simple protocol, network, and graphics problems that could be encountered on much larger projects in industry after graduation.
- Both midterm and final project ensure the student is able to work with many of the advanced features of the java language. Many of the concepts learned with java are applicable in other languages.

Prepared by: John Franco, Ph.D. **Date: June 2009**

Estimated CSAB Category Content:

	Core	*Advanced*		*Core*	*Advanced*
Data Structures:	4 hours		*Computer Org. & Arch.*	0 hours	
Algorithms & Software Design	9 hours		*Concepts of Progrmmng. Languages*	17 hours	

Oral and Written Communications:

Oral and written communication is restricted to understanding the specification of a moderately large project. Communication focuses on expression of creative use of modern programming language facilities to solve complex problems.

Social and Ethical Issues (topics covered):

No ethical issues are discussed in this course

Problem Analysis (common experiences);

Moderately complex problems are offered as homework and the final project. Successful completion of these assignments requires an analysis of the problem as well as supporting code (for example, the final project server). Students work in groups of 2-3 when developing solutions to assignments.

Solution Design (common experiences):

Creative use of advanced concepts such as reflection, threads, custom class loaders, exceptions, sockets, RMI, etc. to implement solutions to problems in computer security, data driven solutions to problems (for example, Hamming's problem extended), graphics problems, and more. Basic knowledge of software design principles is assumed as a prerequisite

U. of CINCINNATI DEPARTMENT COMPUTER SCIENCE JUNE 2009

Faculty Course Feedback

Title: Adv. Programming Concepts **Course:** 20-CS-694 **Professor:** Franco

1. For each outcome addressed by this course, please indicate how it was assessed during the quarter, and to what degree (*high*, *medium*, *low*, or *some* (where *some* indicates that it applies to some students, but not necessarily all)). Assessment includes test questions, homework assignments, project assignments, oral reports, lecture and/or discussion. Please attach at least one example (e.g., assignment, test question, handout) to illustrate this for the course file.

c. an ability to design a system, component, or process to meet desired needs.

A major project, taking 4 weeks of preparation, is required of the students. The project brings together all aspects of the course, especially graphics, networking, and security. The goal is to design a system to seek out running code on several machines, owned by fellow students, and attempt to steal and/or mangle data (wealth) from/within them. Desired needs are specified completely in the attached project and protocol specifications which the students see at the beginning of the quarter.

e. an ability to identify, formulate, and solve problems that arise in their discipline and its applications.

Numerous common problems involving client/server, peer to peer, remote method invocation, class encryption and customized class loaders are either assigned as homework, appear on the midterm exam, or appear in class notes. The project, mentioned above, requires considerations learned from solving these problems. Students need to make their data secure to get a good grade. One student is given a sniffer and eavesdrops on any communication from a class member (this is not told to any other students). Most students are extremely surprised at finding how important security is and how easy it is for an attacker to steal, mangle, or destroy data when protection is not clearly thought out. Solving the problem of security is a big challenge that students must complete successfully to get a good grade.

k. an ability to use the techniques, skills, and modern tools of the discipline necessary for professional practice.

Quite a few advanced programming concepts were presented using tools supplied by the Java language including threads, exceptions, networking, and reflection, among others. Concepts studied include Streams (see example attached for details on objectives), co-routining, semiphores, network security, cryptography, remote method invocation, graphics and animation, data base manipulation using SQL commands via Java classes, and communication protocols.

2. In what ways can the course be improved in the future?

The course must be revamped frequently to keep it up to date. Addition of functional programming techniques, perhaps through Haskell, is a possibility for the future.

3. Please indicate whether the outcomes should be adjusted.

a. Which outcomes should be deleted?
None that I can tell

b. Which additional outcomes should be added? (consult the attached list of outcomes a-k)
None that I can tell