CIS*3190 Outline

Introduction / Welcome

Welcome to CIS*3190, a course which looks at legacy software systems, and in particular various older programming languages, each of which still has its place in the wider world of software.

Course Description

This course is an introduction to legacy software systems used in business, manufacturing, and engineering. Topics include COBOL programming, mainframe systems, and integration of legacy systems with contemporary computing systems. (Offered through Distance Education format only.)

Prerequisite(s): CIS*2500 or work experience in a related field.

About the Instructor

Dr. Wirth is an assoc. professor in the School of Computer Science. His interests lie in programming languages, history of computer science, usability of everyday things, image processing, and computer science pedagogy.

Course Objectives

The objective of this course is to explore three programming languages which symbolize legacy programming languages, and look at their context in modern software.

1. Identify what legacy programs are, why they exist and the difficulties that result from them.
2. Demonstrate a basic level of understanding and application related to legacy code in Fortran, Ada and Cobol.
3. Analyze, translate and modernize, i.e. re-engineer, legacy code for Cobol and Fortran.
4. Be able to review a program written in a legacy language such as Fortran or Cobol, identify legacy and redundant features, and update the program in a newer dialect of the language.
5. Obtain an understanding of the process of program re-engineering.

Resources
Required: Supplied within course notes.
Other:

Learning Strategies

Students solve a series of problems related to legacy programming utilizing information on a variety of programming languages**.**

Course Structure

This course is divided into 12 units, one per week.

Week 01: Unit 01: Introduction to legacy software.

Week 02: Unit 02: Introductory Fortran: Background and elementary language structures (e.g. data, operators, decision statements, I/O, math).

Week 03: Unit 03: Intermediate Fortran: Advanced programming structures (e.g. loops, arrays, subprograms).

Week 04: Unit 04: Modernizing Fortran: Converting programs in old dialects to newer versions of Fortran.

Week 05: Unit 05: Introductory Ada: Background and elementary language structures (e.g. data, operators, decision statements, loops, I/O, math).

Week 06: Unit 06: Intermediate Ada: Intermediate programming structures of Ada (e.g. strings, arrays, subprograms, packages).

Week 07: Unit 07: Advanced Ada: Advanced programming structures of Ada (e.g. exceptions, overloading, files, generics)

Week 08: Unit 08: Introductory Cobol: Background and elementary language structures (e.g. program structure, philosophy of Cobol, data, operators, I/O, math).

Week 09: Unit 09: Intermediate Cobol: Intermediate programming structures of Cobol (e.g. records, decision statements, loops).

Week 10: Unit 10: Modernizing Cobol: Converting programs in old dialects to newer versions of Cobol (e.g. redundant and legacy structures).

Week 11: Unit 11: Case Studies: The Millennium Bug (Y2K) and a look at various real legacy systems.

Week 12: Unit 12: Software Archeology and Re-engineering Legacy Code

Evaluation

Assignment	% of Grade	Due Date
Assignment #1 Legacy Fortran	**25%**	Feb. 3rd, 2017
Assignment #2 Programming in Ada	**25%**	Mar. 3rd, 2017
Assignment #3 Cobol Re-engineering	**20%**	Mar. 24th, 2017
Assignment #4 Legacy Software	**30%**	April 7th, 2017
TOTAL	100%	

Assignment details can be found under the Assignments link in the top navigation bar.

Please see the Schedule in the top navigation bar for exact due dates.

Discussion Use and Expectations

1. **General Discussion**: Use this conference for course-related inquiries. I will always read this Discussion. This is where you go to ask me questions.

2. **Assignment Q&A**: This series of discussion groups (one for each assignment) are used to ask questions regarding the first assignment. I will always read these Discussions, but others may also provide answers to questions. You can share relevant information here pertaining to quirks with any of the compilers, or language features etc.

3. **The Lounge**: Relax, chat and talk about non-course related things.

4. **Introductions**: Here you will introduce yourself to the class. Tell us what you hope to learn by taking this course and anything else that you are comfortable sharing. This discussion is public to all students in the class.

5. **Help**: Technical help staff ready to answer your technical questions about the learning platform will monitor this discussion on a daily basis.

6. **Test**: Use this discussion to learn about the course Discussion system, and to practice sending and receiving messages. Test is a practice discussion common to all web-based courses running this semester.

Remember that non-verbal communication is missing online and give your classmates the benefit of the doubt when interpreting their messages. DO NOT post racist, sexist, homophobic, or other remarks that may cause offense. Please keep in mind that each of the conferences is a public place. Anyone with access to the course website and Discussion area can see your messages. Please use e-mail to me for personal communications.

Policies

Academic Misconduct

Plagiarism

Although students are encouraged to share thoughts and ideas while studying for the course, all material submitted for grading must be each student's own work. For clarification on Academic Misconduct, please refer to the University of Guelph's Undergraduate Calendar or the Open Learning Program Student Handbook . The Learning Commons provides some excellent insight into academic misconduct and plagiarism. Please see General Resources under the Resources link in the top navigation bar.

Acceptable Use

The University of Guelph has an Acceptable Use policy which you are expected to adhere to. Please see this policy under General Resources in the Resources tab in the top navigation bar under the heading **Technical**.

Instructor Contact Information

Michael Wirth
E-mail: mwirth@uoguelph.ca
Telephone: (519) 824-4120 ext. 53889
Office:

Technical Support
Office of Open Learning
University of Guelph
(519) 824-4120 ext. 56939
Toll Free (Can & US) 1-866-275-1478
help@open.uoguelph.ca

Distance Education
University of Guelph
(519) 824-4120 ext. 55000
(519) 824-1112 (fax)
info@open.uoguelph.ca