# CS591-ECE595 (061-068)
# Multiparadigm Programming

## *Presentation and Syllabus*

Manuel Hermenegildo
herme@unm.edu
Room ME 407

*Departments of Computer Science and*
*Electrical and Computer Engineering*
*U. of New Mexico*
*Albuquerque, NM – USA*

**Fall 2005**

**Note:** Course transparencies (including the information on systems, etc. in this set) may change during the term. If you are looking at a printed version it is always advisable to check the on-line version for changes at `http://www.unm.edu/~herme/courses/CS591-ECE595-F05`

## Course Description – Rationale

Functional, Logic, and Constraint Programming languages have proved very successful for developing complex applications ranging from electronic circuit design to machine learning, complex scheduling problems, agents, or the semantic web.

Recently, multiparadigm programming systems have been developed that allow the use of features of all these programming paradigms in a combined way, and some include also features from imperative, object-oriented programming.

Also, significant progress has been made in the context of these languages for a) automatic verification and debugging, and b) developing concurrent applications and for the automatic exploitation of parallelism and resource control in high-performance computing systems.

## Course Description – Objectives

The **objectives** of the course include:

- Becoming familiar with the logic programming, constraint programming (and also functional programming) paradigms, and, specially, their combination.
- Acquiring significant practical experience with multi-paradigm programming environments that support at the same time these three models of programming, as well as including features from imperative- and object-oriented programming.
- Becoming familiar with the techniques used in the context of these paradigms for:
  - Automatic verification and debugging.
  - Automatic exploitation of parallelism and resource control in high-performance computing systems.

It is fundamentally a practical programming course, involving programming exercises with a multiparadigm programming system, but also theoretical foundations will be addressed where relevant.



## Course Description – Prerequisites / Emphasis

- **Prerequisites:**
  - Some familiarity with functional programming and proficiency in imperative and object-oriented programming is assumed.
  - Knowledge of Prolog and proficiency in functional programming is useful but not strictly required.
- **Emphasis:**

  Because familiarity with functional, imperative, and object-oriented programming is assumed, the course concentrates mainly on the other programming paradigms that the student may be less familiar with (i.e., logic- and constraint- programming) in order to then study their combined use.
- The presentation will be adapted as much as possible to student backgrounds.

## General Information

| | |
|---|---|
| **Instructor:** | Manuel Hermenegildo |
| **Office:** | ME 407 |
| **Class time:** | Tuesday and Thursday 9:30–10:45 |
| **Class venue:** | Woodward 149 |
| **Office hours:** | Tuesday and Thursday 2:00-4:00 p.m., or by appointment. |
| **email:** | herme@unm.edu |

- Information on the course available on-line via WWW
  (for reading on-line or printing out)
  - Transparencies for all lectures (including these) in several formats: *pdf*, *postscript*, and *html*.
  - Access to mailing list archive.

  Follow the CS591-ECE595 link from: `http://www.unm.edu/~herme/`
- Standard "mailman"-based course mailing list:
  - Sign up at: `https://www.cs.unm.edu/cgi-bin/mailman/listinfo/cs591`
  - Send messages to: `Cs591@cs.unm.edu`



## Assignments and grading

- (45%) Homework assignments.

  Topics:

  1. Logic Programming.
  2. Combined Functional Programming and LP.
  3. Basic constraint programming.
  4. Advanced constraint programming (+FP and LP).
- (35%) An individual term project, including in-class presentation (more details later, topic will be of your choice: watch out in class for ideas).
- (20%) Short midterm and final, in-class, individual tests.

## Assignments and grading (Contd.)

- Homework and programming assignment hand-in:

  Homework assignments are due at midnight of the due date assigned. Late homework will be given quickly diminishing credit as the deadline passes. Hand-in mechanisms (usually electronic) will be specified with each assignment.

- Academic honesty:

  You are welcome to *discuss* homework with your classmates and the instructors. You must *do* and *write up* your own solution. You cannot look at or copy someone else's solution and you also cannot provide your solution to another student. You are responsible for protecting your homework from unauthorized access. You should ask the instructor promptly if you lack the background to solve a question or find it too difficult.



## Other

- Class attendance is mandatory (can be done by ITV, etc.).
- UNM statement of compliance with ADA:

  Qualified students with disabilities needing appropriate academic adjustments should contact the instructor as soon as possible to ensure their needs are met in a timely manner. Handouts are available in alternative accessible formats upon request.

## Course Outline

- Introduction and Motivation.
- Logic Programming.
  - The (ISO-)Prolog programming language.
  - Efficient Prolog programming.
- Multi-paradigm Programming:
  - Syntactic/semantic extensions.
  - Combining logic programming, functional programming, and objects.
  - Higher order, records.
- Adding Constraint Programming.
  - CLP(R). Finite domains. Current CLP systems. Combined programming.
- Advanced topics:
  - Static debugging and verification.
  - Automatic program parallelization.



## Programming Environment

- Programming environment used in the course: 
  - Multi-paradigm programming environment supporting:
    - Logic programming
    - Functional programming
    - Constraint programming
    - (A form of) Imperative, object-oriented programming

    (allows following the course with a single system)
- Installing/Using Ciao:
  - Freely available from: `http://www.ciaohome.org`
  - You can easily install it in your machine (Linux, Unix, Windows, & Mac).
  - If needed, it can be installed on the CS/ECE Linux machines, and/or on the ECE-215 Windows machines.
- Use of the (`emacs`-based) programming environment highly recommended (syntax highlighting, auto-documenter, multiple debuggers, assertions, etc.).

## Bibliography

- References for LP and CLP:
  - ◇ “The Art of Prolog” (Second edition), Sterling & Shapiro, MIT Press, 1994.
  - ◇ “Programming with Constraints: An Introduction”, Marriott & Stuckey, MIT Press, 1998.
- More advanced LP/CLP references:
  - ◇ “Warren’s Abstract Machine: A Tutorial Reconstruction,” H. Ait-Kaci, MIT Press, 1991.
  - ◇ “Parallel Execution of Prolog Programs: a Survey,” G. Gupta, E. Pontelli, K. Ali, M. Carlsson, and M. Hermenegildo. ACM Transactions on Programming Languages and Systems, ACM Press, 23(4), pages 472–602, 2001.
  - ◇ “Special Issue on Ten Years of Logic Programming,” The Journal of Logic Programming, Volumes 19/20, North Holland, 1994.



## Full info at:

`http://www.unm.edu/~herme`

`http://www.unm.edu/~herme/courses/CS591-ECE595-F05`