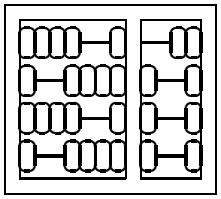

Instituto de Computação
Unicamp

MO-346

Paradigmas de Linguagens de Programação

2º Semestre de 2013

Programa do Curso

1 Resumo da Disciplina

Ementa: Visão comparativa de paradigmas de programação. Programação funcional, lógica.

2 Aulas e Atendimento

	Dia	Horário	Sala
Aulas	3ª	21:00-22:40	CB 08
	5ª	19:00–20:40	CB 06
Atendimento	3ª	16:00-16:50	IC 12
Professor	Siome Goldenstein		
PED	Eric Velten de Melo		

Toda a comunicação com o professor e o PED devem ser feitas pelo sistema de mensagens do moodle para garantir pronto atendimento. Além disto, a principal forma de atendimento é através do forum da disciplina, onde outros alunos podem participar nas discussões e também oferecer respostas.

3 Critério de Avaliação

A disciplina este semestre utilizará fortemente o Moodle para coordenação, divulgação de informações e entrega de trabalhos. É responsabilidade de cada aluno verificar que possui o devido acesso ao sistema no início do semestre. A autenticação é feita pelo sistema do CCUEC utilizando o LDAP da DAC e não é responsabilidade nem do docente nem do Instituto de Computação. Não é permitido fornecer suas credenciais de acesso ao sistema à outro aluno, isto será tratado como fraude.

A avaliação do curso será composta de atividades continuadas (feitas inteiramente pelo sistema do Moodle), diversos projetos de programação em diferentes linguagens, três provas e um exame.

At. Cont.	Projetos	Prova 1	Prova 2	P3
19%	30%	17%	17%	17%

A nota (N) é calculada como uma média ponderada

$$N = 0.19 * AC + 0.3 * Prj + 0.17 * P1 + 0.17 * P2 + 0.17 * P3,$$

e as datas das provas e exame são

Prova 1	19/09/2013
Prova 2	17/10/2013
Prova 3	19/11/2013
Exame	10/12/2013

Alunos com média abaixo de 2.5 estão reprovados por nota. Alunos com média maior que 2.5, mas que ou tenham a nota (N) abaixo de 5.0, ou tenham uma das notas parciais (AC, Prj, P1, P2 ou P3) abaixo de 2.5, precisam fazer o exame (E). A formula da nota final (NF) após o exame é

$$NF = \min\left(\max\left(5.0, N\right), \frac{N+E}{2}\right).$$

1. Prazos: As atividades continuadas e projetos não serão aceitos com atraso.
2. Qualquer caso de fraude ou comportamento anti-ético implicará em média zero na disciplina para todos os envolvidos. Isto é critério de avaliação e não penalidade, logo não impede atitudes posteriores por parte de outras instâncias da Universidade.

4 Programa

Histórico de LP. Sintaxe e semântica. Expressões. Variáveis e escopo. Postscript / PDF. Recursão e controle. λ-Calculus e programação funcional. Lisp puro e operadores funcionais. Tipos de dados e inferência. ML. Abstração de controle. Programação lógica. Prolog. Subprogramas. Abstração de dados. Exceções e eventos. Meta programação. Lua.

5 Bibliografia

1. **CPLS** — Concepts of Programming Language, 10e, Sebesta, Pearson 2012.
2. **CPLM** — Concepts of Programming Language, Mitchel, Cambridge U 2002.

Outras Referências

1. **PLAL** — Programming Languages: An Active Learning Approach. Kent D. Lee. Springer, 2008.
2. **SICP** — Structure and Interpretation of Computer Programs. Abelson, Sussman, and Sussman. MIT Press, 1992.
3. **ML** — ML for the Working Programmer. Lawrence Paulson. Cambridge U Press, 1997.
4. **Clocksin** — Programming in Prolog. Clocksin and Mellish. Springer, 2003.
5. **LAMBDA** — An Introduction to Funcional Programming through Lambda Calculus. Greg Michaelson. Dover, 2011.
6. **PLPP** — Programming Languages: Principles and Paradigms. Maurizio Gabbrielli and Simone Martini. Springer, 2010.